PCT WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification 7 : H04N 5/262 — A1

(11) International Publication Number: WO 00/64148

(43) International Publication Date: 26 October 2000 (26.10.00)

(21) International Application Number: PCT/US00/10451

(22) International Filing Date: 17 April 2000 (17.04.00)

(30) Priority Data:
60/129,854 17 April 1999 (17.04.99) US

(71) Applicant *(for all designated States except US)*: PULSENT CORPORATION [US/US]; 1455 McCarthy Boulevard, Milpitas, CA 95035 (US).

(72) Inventors; and
(75) Inventors/Applicants *(for US only)*: PRAKASH, Adityo [IN/US]; 600 Marlin Court, Redwood Shores, CA 94065–1267 (US). PRAKASH, Eniko, F. [RO/US]; 600 Marlin Court, Redwood Shores, CA 94065–1267 (US).

(74) Agents: ALBERT, Philip, H. et al.; Townsend and Townsend and Crew LLP, 8th floor, Two Embarcadero Center, San Francisco, CA 94111 (US).

(81) Designated States: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, CA, CH, CN, CR, CU, CZ, DE, DK, DM, DZ, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, TZ, UA, UG, US, UZ, VN, YU, ZA, ZW, ARIPO patent (GH, GM, KE, LS, MW, SD, SL, SZ, TZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).

Published
With international search report.

(54) Title: METHOD AND APPARATUS FOR EFFICIENT VIDEO PROCESSING

(57) Abstract

A video compression method and apparatus is disclosed. The present invention includes a "smart" or active decoder (Fig. 3) that performs much of the transmission and the instruction burden that would otherwise be required of the encoder, thus greatly reducing the overhead and resulting in a much smaller encoded bitstream. Thus, the corresponding (i.e., compatible) encoder of the present invention can produce an encoded bitstream with a greatly reduced overhead. This is achieved by encoding a reference frame (Fig. 3, element 7) based on the structural information inherent to the image (e.g., image segmentation, geometry, color, and/or brightness), and then predicting other frames relative to the structural information. Typically, the description of a predicted frame would include kinetic information (Fig. 3, element 6) (e.g., segment motion data and/or inexact matches and appearance of new information, and portion of the segment evolution that is captured by motion per se etc.). Because the decoder is capable of independently determining the structural information (and relationships thereamong) underlying the predicted frame, such information need not be explicitly transmitted to the decoder. Rather, the encoder need only send information that the encoder knows the decoder cannot determine on its own.

ENCODER/DECODER SYSTEM

	Encoder		Decoder
1.	Obtain encode, transmit frame	→	Receive Frame
2	Reconstruct Frame		Reconstruct Frame
3	Segmentation		Segmentation
4	Order segments		Order Segments
5	Obtain new image frame		
6	Determine segment motion		
7	Encode motion information		
8	Determine background residue		
9	Predict background residue fill		
10	Determine sufficiency of prediction		
11	Determine local residue		
12	Order local residue locations		
13	Encode residue		
14	Is 2nd frame keyframe, yes, goto 5	→	Receive Keyframe Flag
15	Transmit motion data	→	Receive motion data
16			Determine and order background residue
17			Predict background residue
18	Transmit background residue data	→	Receive additional background residue data
19			Determine and order local segment residues
20	Transmit local segment residue	→	Receive local segment residue
21	Reconstruct 2nd frame		Reconstruct 2nd frame
22	Goto Step 5		Goto step 5

FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AL	Albania	**ES**	Spain	**LS**	Lesotho	**SI**	Slovenia
AM	Armenia	**FI**	Finland	**LT**	Lithuania	**SK**	Slovakia
AT	Austria	**FR**	France	**LU**	Luxembourg	**SN**	Senegal
AU	Australia	**GA**	Gabon	**LV**	Latvia	**SZ**	Swaziland
AZ	Azerbaijan	**GB**	United Kingdom	**MC**	Monaco	**TD**	Chad
BA	Bosnia and Herzegovina	**GE**	Georgia	**MD**	Republic of Moldova	**TG**	Togo
BB	Barbados	**GH**	Ghana	**MG**	Madagascar	**TJ**	Tajikistan
BE	Belgium	**GN**	Guinea	**MK**	The former Yugoslav Republic of Macedonia	**TM**	Turkmenistan
BF	Burkina Faso	**GR**	Greece			**TR**	Turkey
BG	Bulgaria	**HU**	Hungary	**ML**	Mali	**TT**	Trinidad and Tobago
BJ	Benin	**IE**	Ireland	**MN**	Mongolia	**UA**	Ukraine
BR	Brazil	**IL**	Israel	**MR**	Mauritania	**UG**	Uganda
BY	Belarus	**IS**	Iceland	**MW**	Malawi	**US**	United States of America
CA	Canada	**IT**	Italy	**MX**	Mexico	**UZ**	Uzbekistan
CF	Central African Republic	**JP**	Japan	**NE**	Niger	**VN**	Viet Nam
CG	Congo	**KE**	Kenya	**NL**	Netherlands	**YU**	Yugoslavia
CH	Switzerland	**KG**	Kyrgyzstan	**NO**	Norway	**ZW**	Zimbabwe
CI	Côte d'Ivoire	**KP**	Democratic People's Republic of Korea	**NZ**	New Zealand		
CM	Cameroon			**PL**	Poland		
CN	China	**KR**	Republic of Korea	**PT**	Portugal		
CU	Cuba	**KZ**	Kazakstan	**RO**	Romania		
CZ	Czech Republic	**LC**	Saint Lucia	**RU**	Russian Federation		
DE	Germany	**LI**	Liechtenstein	**SD**	Sudan		
DK	Denmark	**LK**	Sri Lanka	**SE**	Sweden		
EE	Estonia	**LR**	Liberia	**SG**	Singapore		

METHOD AND APPARATUS FOR EFFICIENT VIDEO PROCESSING

1. Brief Introduction

The present invention relates to the compression of motion video data, and more particularly for a synchronized encoder and smart decoder system for the efficient transmittal and storage of motion video data. As consumers desire more motion video intensive modes of communications, the limited bandwidth of current transmission modes, such as broadcast, cable, telephone lines, etc. becomes prohibitive. The introductions of the Internet, and the subsequent popularity the world wide web, video conferencing, digital and interactive television require more efficient ways of utilizing existing bandwidth. Further, motion video intensive applications require immense storage capacity. The advent of multi-media capabilities on most computer systems have taxed tradition storage devices such as hard drives, to the limit.

Compression, as used in this patent, is the means by which digital motion video can be represented efficiently and cheaply. The ultimate goal of video compression is to reduce the bitstream, or video information flow, of the motion video sequences as much as possible, while retaining enough information so that the decoder or receiver can reconstruct the video image sequences in a manner adequate for the specific application, such as television, videoconferencing, etc. The benefit of compression is that it allows more information to be transmitted in a given amount of time, or stored in a given storage medium.

Most digital signals contain a substantial amount of redundant, superfluous, information. For example, a stationary video scene produces nearly identical images in each scene. Compression attempts to remove the superfluous information so that the

related image frames can be represented in terms of the previous, thus eliminating the need to transmit the entire scene for each video frame.

2. Previous attempts

There have been numerous attempts at adequately compressing video imagery. These methods generally fall into one of the following two categories: 1) Spatial redundancy reduction, and 2) Temporal redundancy reduction.

2.1 Spatial Redundancy Removal

The first type of video compression focuses on the reduction of spatial redundancy. Spatial redundancy refers to taking advantage of the correlation among neighboring pixels in order to derive a more efficient representation of the important information in an image frame. These methods are more appropriately termed still image compression routines, as they do not attempt to address the issue of temporal, or frame to frame, redundancy, as explained in section 2.2. They work reasonably well on individual video image frames. However, a critical element in video compression is reducing temporal redundancy, in other words, not having to retransmit, store, or otherwise fully represent, information seen in previous frames. Common still image compression schemes include JPEC, Wavelets, and Fractals.

2.1.1 JPEG/DCT based image compression

One of the first commonly used methods of image compression was the DCT, or direct cosine transformation, compression system, which is at the heart of JPEG.

DCT operates by representing each digital image frame as a series of cosine waves or frequencies. Afterwards, the coefficients of the cosine series are quantized. The higher frequency coefficients are quantized more harshly than those of the lower frequencies are. The result of the quantization is large number of zero coefficients, which

can be encoded very efficiently. However, JPEG and similar compression schemes do not address this crucial issue of temporal redundancy.

2.1.2 Wavelets

As a slight improvement to the DCT compression scheme, the wavelet transformation compression scheme was devised. This system is similar to the DCT. The only substantial difference is that the image frame is represented as a series of wavelets, or windowed oscillations, instead of as a series of cosine waves.

2.1.3 Fractals

The goal of fractal compression is to take an image and determine the single function or set of functions, which fully describe the image frame. A fractal is an object that is self-similar at different scales, or resolutions, i.e. no matter what resolution you look at, the object remains the same. Theoretically, fantastic compression ratios could occur as simple equations describe complex images.

Fractal compression is not a viable method of general compression. The high compression rations only work on specially constructed images, and only with considerable help from a person guiding the compression process. Fractal Compression is a computationally intensive process.

2.2 Temporal and Spatial Redundancy Removal

Adequate motion video compression requires reduction of both temporal and spatial redundancies within the sequence of frames that comprise video. Temporal redundancy removal is concerned with the removal from the bitstream, information that had already been coded in previous image frames. Block matching is the basis for most currently used effective means of temporal redundancy removal.

2.2.1 Block Based Motion Estimation

Block Matching is the process by which a block of the image is subdivided into uniform size blocks and each block is tracked from one frame to another and represented by a motion vector instead of having the block re-coded and placed into the bitstream for a second time. Examples of compression routines that use block matching include MPEG, and all its variants.

MPEG operates by performing a still image compression on the first frame and transmitting it. It then divides the same frame into 16 pixel by 16 pixel square blocks and attempts to find each block within the next frame. For each block that still exists in the subsequent frame, MPEG needs only transmit the motion vector, or movement, of the block along with sufficient identifying information. As the block moves from frame to frame, it may not remain the same. The difference is known as the residue. Additionally, as blocks move, previously hidden areas may become visible for the first time. This is also known as the residue. Collectively, the remaining information after the block motion is sent is known as the residue frame, which is coded using JPEG and sent to the receiver to complete the image frame.

Next, the encoder divides the second image frame into blocks and the routine continues until a new keyframe is inserted. A keyframe is an image frame which is completely self-contained, not described in relation to any other image frame.

Although state of the art, block matching is highly inefficient and fails to take advantage of the known general physical characteristics of images. For example, the block method is inherently crude, as the blocks do not have any relationship with real objects in the image. A given block may comprise a part of an object, a whole object, or even multiple dissimilar objects with unrelated motion. In addition, often, neighboring

objects will have similar motion. However, since blocks do not correspond to real objects, block based systems cannot use this information to further reduce the bitstream

Another major limitation of block based matches is the residue frame coding. The residue frame created after block based matching will generally be noisy and patchy and does not lend itself to good compression via standard image compression schemes such as DCT, wavelets, or fractals.

2.3 Alternatives

It is well recognized that the current state of the art needs improvement, specifically the block based method is extremely inefficient and does not produce an optimally compressed bitstream for motion video information. To that end, the latest compression schemes, such as MPEG4 allows for the inclusion of the structural information, if available, of selected items within the frames instead of merely using arbitrary sized blocks. While, some compression gains are achieved, the overhead information is substantially increased because in addition to the motion and residue information these schemes require that the structural or shape information for each item must be sent to the receiver. This is because all current compression schemes use a dumb receiver, one, which is incapable of making determinations for itself.

Additionally, as mentioned above, the current compression methods code the residue frame merely another image frame to be compressed by JPEG, without attempting to determine is more efficient methods are possible.

3. Novel Approaches

This invention represents a novel approach to the problem of video compression. As described above, the goal of video compression is to represent accurately a sequence of video frames with the smallest bitstream, or video information flow. As previously

stated, spatial redundancy reduction methods above are inappropriate for motion video compression. Further, the current temporal and spatial redundancy reduction methods such as MPEG2 waste precious bitstream space by having to transmit a lot of overhead information. This invention solves that problem by using a smart decoder. This smart decoder determines much of the overhead information, thus obviating the necessity of transmitting such information, and therefore reducing the bitstream accordingly.

The smart decoder also makes the same predictions about the subsequent images in the related sequence of images as the encoder. Thus, the encoder can simply send the difference between the prediction and the actual values, thus also reducing the bitstream,

DETAILED DESCRIPTION

1. Introduction/Summary

Compression of digital motion video is the process by which superfluous or redundant information, both spatial and temporal, contained within a sequence of related video frames (frames) is removed. Video compression allows the sequence of frames to be represented by a reduced bitstream, or data flow, while retaining its capacity to be reconstructed in a visually sufficient manner.

Traditional methods of video compression place most of the compression burden, i.e. computational and transmittal, on the encoder, while minimally using the decoder. A tradition video encoder/decoder system requires that the encoder makes all the calculations, inform the decoder of its decisions, then transmit the video data to the encoder along with instructions for reconstruction of each image.

This invention is novel in that it uses a smart decoder to take much of the transmission and instructional burden from the encoder which results in a much smaller

bitstream. Specifically, absent from the bitstream is the information regarding the structural information inherent within the image frame, such as geometry, color, and brightness, which, in a complex frame is a significant amount of video information. Further, absent from the bitstream is information regarding any decision made by the encoder such as segment ordering, segment association and disassociation, etc.

Fig. 1 is an overview drawing of the encoder for use with a compatible decoder as will be described later with respect to Fig. 2. The encoder works as follows:

1. The encoder obtains a reference image frame;
2. The encoder encodes the image frame from step 1;
3. The encoded image from step 2 is reconstructed by the encoder, in the same manner as the decoder will;
4. The encoder segments the reconstructed image from step 4; Alternatively, the encoder segments the original reference image frame from step 1;
5. The segments determined in step 4 are ordered by the encoder, in the same manner as the decoder will;
6. The encoder obtains a new image frame;
7. The motion or kinetic information of each segment, determined in step 4, from the reconstructed, or original image in step 3, to the new image frame in step 6 is determined by motion matching;
8. The encoder encodes the kinetic information;
9. Based on the motion information from step 8, previously hidden regions, also known as the background residue, in the first frame may be exposed in the second frame;

10. The encoder orders the Background residues, in the same manner as the decoder will;
11. The encoder attempts to fill each of the background residues from step 9 and 10.
12. The encoder determines the difference between the predicted fill and the actual fill for each of the background residue areas.
13. The encoder determines the local residue areas in the second image frame, from the segment motion information;
14. The encoder orders the local residues from step 13, in the same manner as the decoder will;
15. The encoder encodes the local residues from step 13.
16. The encoder determines any special instructions associated with the segment information
17. If the image can be reasonably reconstructed primarily from the kinetic information, with assistance from the background residue and the local segment residues, the encoder transmits the following information, and reconstructs the second frame, and continues at step 6:
    a. Flag denoting that the second frame is not a keyframe;
    b. The kinetic information for the segments;
    c. The special instructions for the segments;
    d. The background residue information along with flags denoting coding;
    e. The local residue information along with flags denoting coding;

18. If the image cannot not be reconstructed in relation to the reference frame, the image is encoded as a flag transmitted to inform the decoder, and the encoder continues at step 2.

Fig 2 is an overview drawing of the decoder system with a compatible encoder as described in Fig 1.. The decoder system works as follows:

1. The decoder receives a first encoded image frame from step 3 of the encoder description;
2. The encoded image frame from step 1 is reconstructed by the decoder in the same manner as the encoder;
3. The reconstructed image frame from step 2 is segmented by the decoder. Alternatively, the reconstructed image frame is not segmented by the decoder
4. The decoder receives a flag from the encoder stating whether the second frame from step 19 and 20 of the encoder description is a keyframe, i.e. not represented in relation to any other frame. If so, then the decoder returns to step 1.
5. The decoder receives motion information regarding the segments determined in step 3 from the encoder;
6. The decoder begins to reconstruct a subsequent image frame using the segments obtained in step 3 and motion information obtained in step 4;
7. Based on the motion information from step 4 regarding the segments determined in step 3, the decoder determines where areas, previously hidden, are now revealed, also known as the background residue;
8. The previously background residue locations from step 6 are ordered in the same manner as in the encoder;

9. The decoder attempts to fill the background residue locations from step 6;
10. The decoder receives additional background residue information plus flags denoting the coding method for the additional background residue information from step 8 from the encoder;
11. The decoder decodes the additional background residue information;
12. The computed background residue information and the added background residue information is added to the second image frame.
13. Based on the motion information from step 4 regarding the segments determined in step 3, the decoder determines the location of the local segment residues.
14. The local segment residue locations are ordered in the same manner as the encoder does;
15. The decoder receives coded local segment residue information plus flags denoting the coding method for each local segment residue location;
16. The decoder decodes the local segment residue information;
17. The decoded local segment residue information is added to the second frame.
18. The encoder receives the special instructions, if any, for each segments
19. Reconstruction of the second frame is complete;
20. If there are more frames, the routine continues at step 4

Fig 3 is an overview drawing of the encoder/smart decoder system. The encoder/smart decoder system works as follows:

1. The encoder obtains, encodes and transmits the reference frame;
2. The reference frame from step 2 is reconstructed by both encoder and decoder;

3. Identical segments in the reference frame are determined by both encoder and decoder;
4. The segments from step 3, are ordered in the same way by both the encoder and decoder;
5. The encoder obtains a new image frame;
6. The encoder determines the motion of segments from step 3 by means of motion matching frame from step 5;
7. The encoder encodes motion information;
8. Based on motion information from step 7, the encoder determines previously hidden areas, also known as background residue, which is now exposed in the second frame.
9. The encoder attempts to mathematical predict the image at the background residue regions.
10. The encoder determines if the mathematical prediction was good based upon the difference between the guess and the prediction. The encoder computes additional background residue if necessary.
11. Based on segment information in step 3, and the motion information from step 7, the encoder determines structural information for the local segment residues;
12. Structural information for the local residues from step 11 are ordered by the decoder.
13. Based on the structural information from step 12, regarding the local residues, the encoder encodes the local segment residues.

14. The encoder determines if based upon the kinetic information of the segments, if the second frame should be coded in reference to the first frame. If not, it is coded as a keyframe and the routine begins at step 1.

15. The decoder receives the segment kinetic information from the encoder in step 7.

16. The decoder determines and orders the same background residue at the encoder did in step 8.

17. The decoder makes the identical guess as to the structure of the background residue as the encoder did in step

18. The decoder determines and orders the same local segment residues as determines in step 11 and 12.

19. The decoder receives the local segment residues information from the encoder and flags denoting the coding scheme.

20. The decoder receives the additional background residue information from the encoder.

21. The encoder receives the special information, if any, regarding each segment.

22. Based upon the kinetic information, the local segment residues, and the background residues, both the encoder and decoder identically reconstruct the second frame.

23. The second frame is now the reference frame and the process continues at step 5.

ENCODER WRITE-UP

2. Reference Frame Transmission

Referring to Fig 4, the encoder receives the reference frame, in this case, a picture of an automobile moving left to right with a mountain in the background. The reference frame generally refers to the frame which any other frame is described in relation to.

Fig. 5 is the part of the flow diagram illustrating the procedure by which the encoder initially processes the reference frame. Step 110 begins the process, specifically, the encoder receives the picture described in Fig.4. At step 120, the encoder encodes Fig 4, into a video format, and transmits it to the receptor at step 130. The encoder reconstructs the encoded frame at step 140.

3. Segmentation

Segmentation is the process by which a digital image is subdivided into its component parts, i.e. segments, where each segment represents an area bounded by a radical or sharp change in values within the image.

Persons well versed in the art of computer vision will be aware that segmentation can be done in a plurality of ways. One such way is the watershed method where each pixel is connected to every other pixel in the image frame. As seen in Fig 6, the watershed method segments the image by disconnecting pixels based upon a variety of algorithms. The remaining connected pixels belong to the same segment.

Referring to Fig. 6, At step 210, the encoder segments the reconstructed reference frame to determine the inherent structural features of the image. Alternatively, at step 210, the encoder segments the original image frame for the same purpose. The encoder determines that the segments of Fig. 2 are the car, the wheels, the windows, the street, the sun, and the background. At step 220, the encoder orders the segments based upon a pre-determined criteria and marks them Segments 1 through 8, respectively, as seen in Fig 7.

Segmentation permits the encoder to perform efficient motion matching, motion prediction, and efficient residue coding as explained further in this description.

4. Kinetic Information

Once segmentation has been accomplished, the encoder encodes the kinetic or motion information regarding the movement of each segment.

The kinetic information is determined through a process known as motion matching. Motion matching is the procedure of matching similar regions, often segments, from the first frame to the second frame. At each pixel within a digital image frame, an image is represented by numerical value. Matching occurs when a region in the first frame has identical or near identical pixel values with a region in the second frame.

Generally speaking, a segment is matched with a segment in another frame when the absolute value of the difference in pixel values between the segments is below a pre-determined threshold. While the absolute value of the pixel difference is often used to because it is simple and accounts for negative numbers any number of function would suffice.

In Fig 7a, we see an example of motion matching of a soccer ball between frames 1 and 2. In frame 1, we have a soccer ball, with black and white squares. In frame 2, we have a brownish orange basketball next to the soccer ball. Subtraction of the pixels values contained within the basketball in frame 2 from the soccer ball in frame 1 yield a relatively arbitrary set of non-zero differences. Thus the soccer ball and basketball will not be matched. However, subtraction of the soccer ball in frame 2 from the soccer ball in frame 1 yields a set of mostly zero and close to zero values. Thus the two soccer balls would be considered matched.

The kinetic information transmitted to the decoder can be reduced if related segments can be considered as single groups so that the encoder only needs to transmit one main representative motion vector to the decoder along with motion vector offsets to represent the individual motion of each segment within the group. Grouping is possible if there is previous kinetic information about the segments or if there is multi-scale information about the segments. Multi-scaling will be explained in section 4.2 of the encoder discussion.

Referring to Fig 8, at step 310, the encoder determines if the first frame is a keyframe, i.e. not described in relation to other frames. If the first frame is a keyframe, then there isn't any previous kinetic information and grouping is only possible if there is multi-scale information regarding the image frame. However, if the first frame is not a keyframe, then there will be some previous kinetic information to group segments. Therefore, if the first frame is not a keyframe, step 320, will execute the motion grouping routine, described here as section 4.1.

However, if the first frame is a keyframe, then step 310, goes to step 330, where the encoder determines if there is any multi-scale information available to it. If there is, then step 340 executes the Multi-scaling routine in section 4.2, otherwise at step 350, the encoder decides not to group any segments.

If the first frame is a keyframe, and thus previous kinetic information is not available, and there is no multi-scale information available either, the encoder cannot group the segments and then, at step 350, encoder determines that it cannot group any segments together.

4.1 Motion Vector Grouping

Motion vector grouping only occurs when there is previous motion information so that the encoder can determine which segments to associate. Motion vector grouping begins at step 510 in Fig 10, where the previous motion vector of each segment is considered. Segments which exhibit similar motion vectors are grouped together at step 520. At step 530, the motion vector for the group is determined by combining the motion vectors within the groups. Thus, for each segment within the group, only the motion vector difference, i.e. the difference between the segment's motion vector and the characteristic motion vector will be eventually transmitted. (See step 540) One example of a characteristic motion vector would be an average motion vector.

At step 550, the encoder orders the groups. However, before the motion information can be transmitted, further reduction might occur through motion prediction at step 555, described here in section 4.1.1. Once the motion information is determined it is stored at step 560.

4.1.1 Motion Prediction

Referring to Fig 11, at step 610, the encoder considers a segment. At step 620, the encoder determines if there is previous motion information for the segment so that its motion can be predicted. If there isn't any previous motion information, the encoder chooses the next segment and continues.

If there is previous motion information the encoder predicts the motion of the segment at step 630 and compares its prediction to the actual motion of the segment at step 640. The motion vector offset is initially predicted at step 650 as a function of the actual and predicted motion vectors. An example of a motion vector calculation would be the difference between the actual and predicted motion vectors. At step 660 the

encoder makes the final calculation for the motion vector offset. An example of the final motion vector calculation could be the difference between the initial motion vector and the characteristic motion vector.

At step 670, the encoder determines if there are any more segments, if so, then at step 680, the encoder considers the next segment and continues at step 620. Otherwise the prediction routine ends.

4.2 Multi-Scale Grouping

Multi-scaling grouping is an alternative to grouping segments by previous motion. Moreover, multi-scaling in may used in conjunction with motion grouping. Multi-scaling is the process of creating lower resolution versions of an image. An example of creating multiple scales is through the repeated application of a smoothing function. The result of creating lower resolution images is that as the resolution decreases, only larger, more dominant features remain visible. Thus for example, the stitching on a football may become invisible at lower resolutions, yet the football itself remains discernible

An example of the multi-scale processes is as follows: referring to Fig. 9 at step 410, the encoder considers the coarsest image scale (i.e. lowest resolution) for the frame and at step 420 determines which segments have remained visible. The coarsest image scale is used because at that point, only the absolute largest, most dominant features remain, usually corresponding to the outline of major objects remain visible. While smaller, less dominant segments are no longer discernible at the lower resolutions. At step 430, invisible segments which are wholly contained within a given visible segment are associated with the segment and considered one group. This is because the smaller, now invisible segments are often share a relationship with the larger object and will likely

have similar kinetic information. A decision is made at step 440. If there are more visible segments, at step 450, the encoder considers the next segment and continues at step 430. Otherwise the Multi-scaling grouping process ceases.

5. Residue Coding

Referring to Figs. 12-15, the residue is the portion of the image left over after the structural information has been moved. Residue falls under two classifications; new information and local residues.

5.1 New information

As shown in Fig. 12, as the segment moves, previously hidden or obstructed areas may become visible for the first time. In Fig. 12, three regions become visible as the car moves. They are the area behind the back of the car and the two areas behind the wheels. These are marked regions 1 through 3, respectively. Referring to Fig 13, at step 710, the encoder determines where the previously obstructed image regions occur. At step 720, the encoder orders the region using a predetermined ordering system. Using the information surrounding the regions, the encoder makes a mathematical guess as to the structure of the regions. Yet, the encoder also knows precisely what images were revealed at these regions. Thus at Step 740, the encoder considers a region and determines if the mathematical prediction was sufficient by comparing the guess with the actual image. If the prediction was not close, at step 770, the encoder will encode the region or the difference and store the encoded information with a flag denoting the coding mechanism. Otherwise, if the guess was close enough, the encoder stores a flag denoting that fact at step 745.

At step 750, the encoder determines if there are any more newly unobstructed regions. If so the next region is considered and the routine continues at step 730, else it ceases at step 799.

5.2 Local residues

Referring to Fig. 14, the local residue is the portion of the image in the neighborhood of a segment, left over after the segments have been moved, i.e. the car and mountain appear smaller in the subsequent frame. The structure of the residue will depend on how different the new segments are from the previous segments. It may be a well-defined region, or set of regions, or it may be patchy. Different types of coding methods are ideal for different types of local residue. Since the decoder knows the segment motion, it knows where most of the local residues will be located.

Referring to Fig 15, at step 810, the encoder determines the locations of the local residues and orders the regions where the local residues occurs using a pre-determined ordering scheme at section 820. At step 830, the encoder considers the first local residue, and makes a decision as the most efficient method of coding it and encodes it at step 840. The encoder stores a flag denoting the coding mechanism as well as the coded residue at step 850. If there are more local residue locations, step 860 will consider the next local residue location and continue at step 840, otherwise the at step 870, the encoder executes the keyframe routine at Fig 15a, step 880.

Referring to Fig 15a, at step 880, the encoder determines if the second frame should be coded as a keyframe. If yes, then step 885, the encoder discards the kinetic information, the background residue, and the local segment residues and continues at step 120. Otherwise, the routine transmits the kinetic information, the background residue, and the local segment residues to the decoder at step 890.

6. Special Commands

The encoder transmits embedded commands and instructions regarding each segment into the bitstream as necessary. Examples of these commands include, but are not limited to, getting static web pages, obtaining another video bitstream, waiting for text, etc.

The encoder can embed these commands at any point within the bitstream subsequent to the decoder ordering the segments. Fig 14a, is an example of one point where the commands are be embedded within the data stream.

Referring to Fig 14a, at step 1610, the encoder considers the first segment. At step 1620, it transmits a special instruction. At step 1630, the encoder determines if there are any special instructions for the segment. If yes, then at step 1640, the instructions are transmitted to the decoder and at step 1650 the encoder determines if there are any more segments. If there are no special instructions associated with the segment, the encoder proceeds directly to step 1650. If there are more segments, at step 1660, the encoder considers the next segments are continues to step 1620, otherwise the routine ends at step 1699.

DECODER DESCRIPTION

2. Reference Frame Reception

Referring to Fig 16, the decoder receives the encoded reference frame of a picture of an automobile moving left to right with a mountain in the background (See Fig. 4). The reference frame generally refers to the frame which other, subsequent frames are described in relation to.

Fig. 16 illustrates the flow diagram of the above process. Step 910 begins the process where the decoder receives an encoded image frame. At step 920, the decoder reconstructs the encoded image frame.

At step 930, the decoder receives a keyframe flag. This flag denotes whether the second frame is a keyframe or can it be reconstructed from the kinetic and residue information. If the second frame is a keyframe, then the decoder returns to step 910, where it received the keyframe as a first frame, otherwise the routine continues.

3. Segmentation

As previously described, segmentation is the process by which a digital image is subdivided into its components parts, i.e. segments, where each segment represents an area bounded by a radical or sharp change in values within the image.

Referring to Fig 17, at step 1010, the decoder segments the reconstructed reference frame to determine the inherent structural features of the image. The decoder determines that the segments in Fig. 4 are the car, the wheels, the doors, the windows, the street, the mountain and the background. At step 1020, the decoder will order the segments based upon the same predetermined criteria as the encoder and mark the segments as 1 through 10 as seen in Fig 7.

4. Kinetic Information

Once segmentation has been accomplished, the decoder receives a keyframe flag from the encoder. This flag tells the encoder if the first frame is a keyframe. The decoder receives the kinetic information regarding the movement of each segment. The kinetic information tells the decoder the position of the segment in the new frame relative to its position in the previous frame. The kinetic information is reduced if the segments with related motion can be grouped together and represented by one motion vector. The

kinetic information received by the decoder depends on several factors: to wit; 1) the reference frame is a key frame, and 2) if not, is multi-scaling information available.

Referring to Fig. 18, at step 1110, the decoder determines if the reference frame is a keyframe, i.e. a frame not defined in relation to any other frame. If so, then there is no previous motion information for potential grouping of segments, therefore the decoder attempts to use multi-scale information for segment grouping, if available. At step 1120, the decoder determines if there is multi-scale information available.. If the first frame is a keyframe and there is multi-scale information available to the decoder, the decoder will initially group related segments together using the multi-scale routine executed at step 1130, and described in section 4.1 of the description. Conversely, if there is no multi-scale information available for the first frame, then at step 1150, the motion vectors are transmitted by the encoder and received by the decoder.

However, at step 1110, if the decoder determines that the first frame is not the keyframe, then it executes the motion grouping routine at step 1140, and described in section 4.2. Alternatively, it may use the multi-scale grouping described in step 4.1

4.1 Multi-Scale Grouping

Multi-scale grouping only occurs when the first frame is a keyframe and there is multi-scale information available to the decoder.

Referring to Fig. 19 at step 1210, the decoder considers the coarsest image scale for the frame and at step 1220 determine which segments have remained visible. At step 1230, invisible segments which are wholly contained within the a given visible segment are associated with the segment. A decision is made at step 1240. If there are more visible segments, at step 1260, the decoder considers the next segment and continues at

step 1230. Otherwise the Multi-scaling grouping process receives the motion vectors and motion vector offsets for the segments then ceases.

4.2 Motion Vector Grouping

Referring to Fig 20, at step 1310, the decoder considers a segment. At step 1320, the encoder determines if there is previous motion information for the segment so that its motion can be predicted. If there isn't any previous motion information, the encoder chooses the next segment and continues.

If there is previous motion information the encoder predicts the motion of the segment at step 1330 and receives the motion vector prediction correction at step 1340.

At step 1350, the encoder determines if there are any more segments, if so, then at step 1360, the encoder considers the next segment and continues at step 1320. Otherwise the prediction routine ends.

5. Residue Coding

The residue is the portion of the image left over after the structural information has been moved. Residue falls under two classifications; background and local residues.

5.1 Background residue

As shown in Fig 12, as the car moves, previously hidden or obstructed areas may become visible for the first time. The decoder knows where these areas are and orders them using a predetermined ordering scheme. In Fig 12. Three regions become unobstructed, specifically, behind the car, and behind the two wheels. These regions are marked Regions 1 through 3, as seen in Fig 12.

Referring to Fig 21, at step 1410, the decoder considers the background residue regions and orders the regions at step 1420. At step 1430, it makes a mathematical prediction on the structure of the first background residue location. At step 1440, the

decoder receives a flag denoting how good the prediction was and if correction is needed. Step 1450 makes a decision, if the prediction is sufficient, the routine continues at step 1470, otherwise at step 1460, receives the encoded region and the flag denoting the coding scheme and reconstructs as necessary. If there are more background residue locations, at step 1470, the decoder, at step 1480, considers the next region and continues at step 1430. Otherwise the decoder goes to step 1490 where reconstruction continues and the process ceases.

5.2 Local residues

Referring to Fig 15, as previously explained, the local segment residue is the portion of the image, in the neighborhood of the segment, left over after the segment has been moved, i.e. the car and the mountain appear smaller in the subsequent frame. Also, as explain before, the structure of the local residue may be varied. The decoder knows that most of the local residues will appear around the segments.

Referring to Fig. 23, at step 1510, the decoder considers the first segment. At step 1520, the decoder receives a flag denoting the coding method and receives the encoded local residue for that segment. Step 1530 determines if there are any more segments and if not end at 1590 where reconstruction concludes. Otherwise at step 1540 the decoder considers the next segment and continues at step 1520. The routine ends at step 1599.

6. Special instructions

In addition to structural information regarding the image frame, the decoder is capable of receiving and executing commands embedded within the bitstream and associated with the various segments. As before, because the encoder and decoder are synchronized and are working with the same reference frame, the encoder is not required to transmit the structural information associated with the commands. The embedded

commands are held in abeyance until a user-driven event, i.e. a mouseclick, occurs. Fig 24 is an example of one potential way to embed the commands.

Referring to Fig 24, at step 1710, the decoder considers the first segment, at step 1720 it received a special instruction flag. The decoder determines, at step 1730, if there are special instructions or commands associated with the segment. If so, the decoder receives the commands at step 1740. At step 1750, the decoder determine if there are any more segments. If there were no special instructions or commands, the decoder goes to step 1750 directly.

If there are more segments, the decoder, at step 1760, considers the next segment and continues at step 1720, otherwise the routine ends at step 1799.

Referring to Fig 25, at step 1810, the decoder determines if the user-driven event has occurred. If it has, the decoder determines which segment the user-driven event refers to at step 1820. At step 1830, the associated command is executed. The decoder proceeds to step 1840. If the user-driven event has not occurred, the routine proceeds directly to step 1840. At step 1840, if the termination command has been sent, the routine exits at step 1899, otherwise the routine continues at step 1810.

6. Reconstruction

The second frame is reconstructed into a video format based upon the kinetic motion of the segments, and local segment residues and the background residues.

Video format

The description in the previous sections titled encoder and decoder description defines a specific new video format.

WHAT IS CLAIMED IS:

1. A method of transmitting video information comprising:

(a) obtaining a first video frame containing image data;

(b) obtaining structural information inherent in said image data;

(c) obtaining a second video frame to be encoded relative to said first video frame;

(d) computing kinetic information for describing said second video frame in terms of said structural information of said first video frame; and

(e) transmitting said kinetic information to a decoder for use in reconstructing said second video frame based on said decoder's generation of said structural information of said first video frame.

ENCODER DESCRIPTION

1	The encoder obtains a reference image frame
2	The encoder encodes the image frame from step 1 and transmits it to the decoder.
3	The encoded image from step 2 is reconstructed by the encoder, in the same **manner as the decoder will;**
4	The encoder segments the reconstructed image from step 4; Alternatively, the **encoder segments the original reference image frame from step 1;**
5	The segments determined in step 4 are ordered by the encoder, in the same **manner as the decoder will;**
6	The encoder obtains a new image frame;
7	The motion or kinetic information of each segment, determined in step 4, from the reconstructed, or original image in step 3, to the new image frame in step 6 is determined by motion matching;
8	The encoder encodes the kinetic information;
9	Based on the motion information from step 8, previously hidden regions, also known as the background residue, in the first frame may be exposed in the second frame;
10	The encoder orders the Background residues, **in the same manner as the decoder will;**
11	The encoder attempts to fill each of the background residues from step 9 and 10;
12	The encoder determines the difference between the predicted fill and the actual fill for each of the background residue areas.
13	The encoder determines the local residue areas in the second image frame, from the segment motion information;
14	The encoder orders the local residues from step 13, **in the same manner as the decoder will;**
15	The encoder encodes the local residues from step 13.
16	If the image can be reasonably reconstructed primarily from the kinetic information, with assistance from the background residue and the local segment residues, the encoder transmits the following information, and reconstructs the second frame, and continues at step 6: a. Flag denoting that the second frame is not a keyframe b. The kinetic information for the segments c. The background residue information along with flags denoting coding The local residue information along with flags denoting coding
17	If the image cannot not be reconstructed in relation to the reference frame, the image is encoded as a flag transmitted to inform the decoder, and the encoder continues at step 2.

Fig 1.

DECODER DESCRIPTION

1	The decoder receives a first encoded image frame from step 3 of the **encoder description**;
2	The encoded image frame from step 1 is reconstructed by the decoder in **the same manner as the encoder**;
3	The reconstructed image frame from step 2 is segmented by the decoder. **Alternatively, the reconstructed image frame is not segmented by the decoder**
4	The decoder receives a flag from the encoder stating whether the second frame from step 19 and 20 of the encoder description is a keyframe, i.e. not represented in relation to any other frame. If so, then the decoder returns to step 1.
5	The decoder receives motion information regarding the segments determined in step 3 **from the encoder;**
6	The decoder begins to reconstruct a subsequent image frame using the segments obtained in step 3 and motion information obtained in step 4;
7	Based on the motion information from step 4 regarding the segments determined in step 3, the decoder determines where areas, previously hidden, are now revealed, **also known as the background residue**;
8	The previously background residue locations from step 6 are ordered in **the same manner as in the encoder**;
9	The decoder attempts to fill the **background residue locations** from step 6;
10	The decoder receives additional background residue information plus flags denoting the coding method for the additional background residue information from step 8 **from the encoder**;
11	The decoder decodes the additional background residue information;
12	The computed background residue information and the added background residue information is added to the second image frame.
13	**Based on the motion information from step 4 regarding the segments determined in step 3, the decoder determines the location of the local segment residues.**
14	The local segment residue locations are ordered in the same manner as the encoder does;
15	The decoder receives coded local segment residue information plus flags denoting the coding method for each local segment residue location;
16	The decoder decodes the local segment residue information;
17	The decoded local segment residue information is added to the second frame.
18	Reconstruction of the second frame is complete;
19	If there are more frames, the routine continues at step 4.

FIG. 2

ENCODER/DECODER SYSTEM

	Encoder		Decoder
1.	Obtain encode, transmit frame	→	Receive Frame
2	Reconstruct Frame		Reconstruct Frame
3	Segmentation		Segmentation
4	Order segments		Order Segments
5	Obtain new image frame		
6	Determine segment motion		
7	Encode motion information		
8	Determine background residue		
9	Predict background residue fill		
10	Determine sufficiency of prediction		
11	Determine local residue		
12	Order local residue locations		
13	Encode residue		
14	Is 2nd frame keyframe, yes, goto 5	→	Receive Keyframe Flag
15	Transmit motion data	→	Receive motion data
16			Determine and order background residue
17			Predict background residue
18	Transmit background residue data	→	Receive additional background residue data
19			Determine and order local segment residues
20	Transmit local segment residue	→	Receive local segment residue
21	Reconstruct 2nd frame		Reconstruct 2nd frame
22	Goto Step 5		Goto step 5

FIG. 3

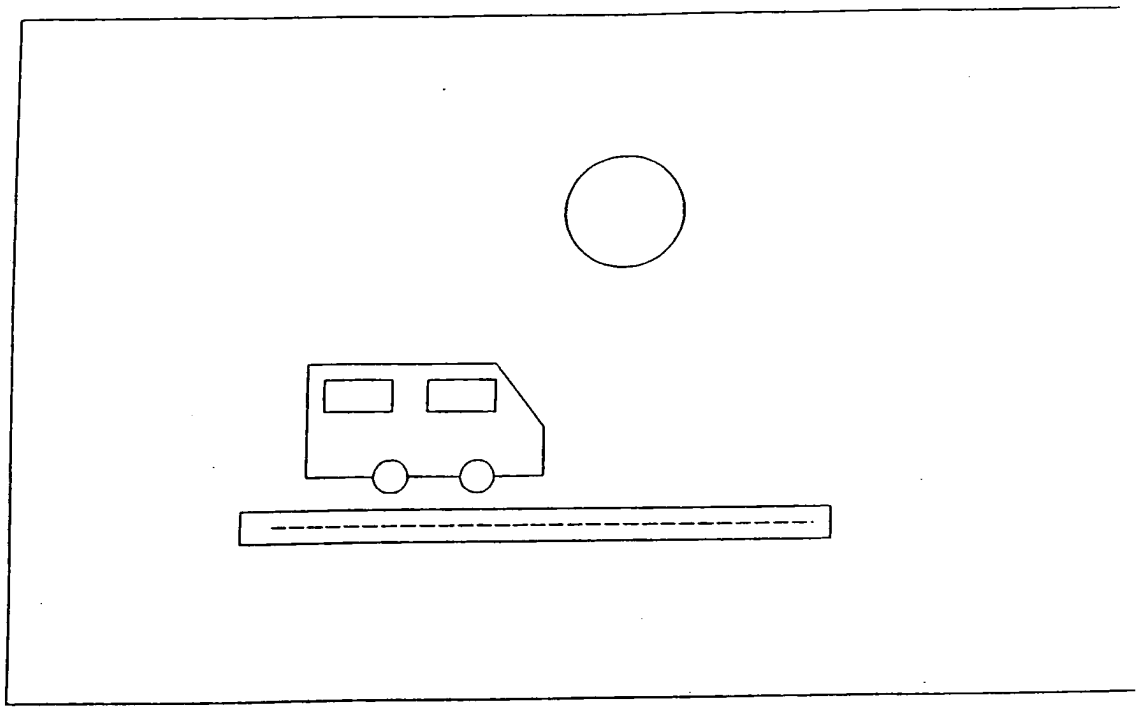

FIG. 4

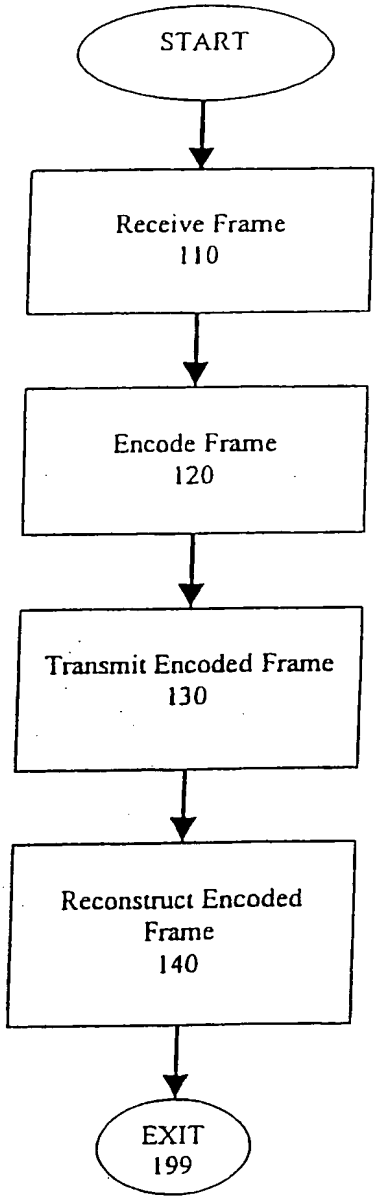
START
Receive Frame
110
Encode Frame
120
Transmit Encoded Frame
130
Reconstruct Encoded
Frame
140
EXIT
199

FIG. 5

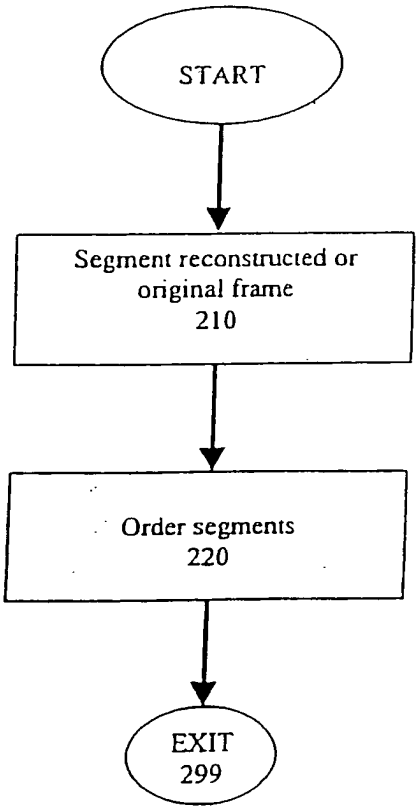
START
Segment reconstructed or original frame 210
Order segments 220
EXIT 299

FIG. 6

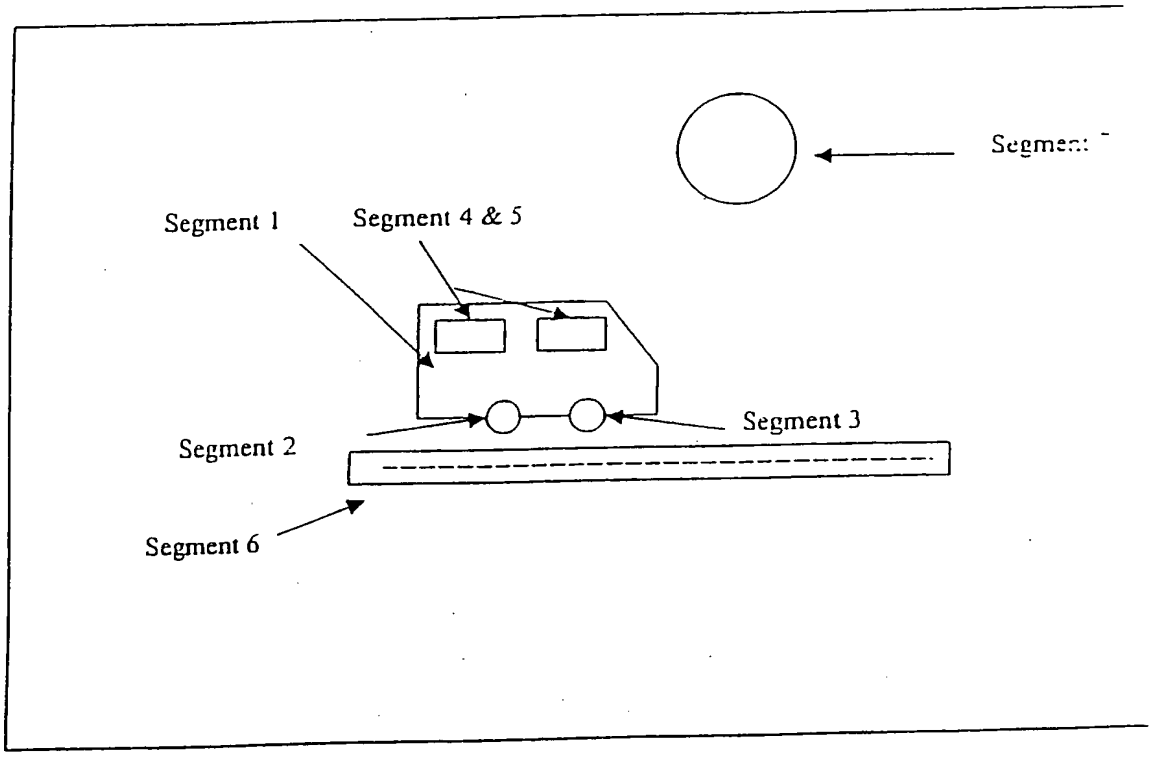

FIG. 7

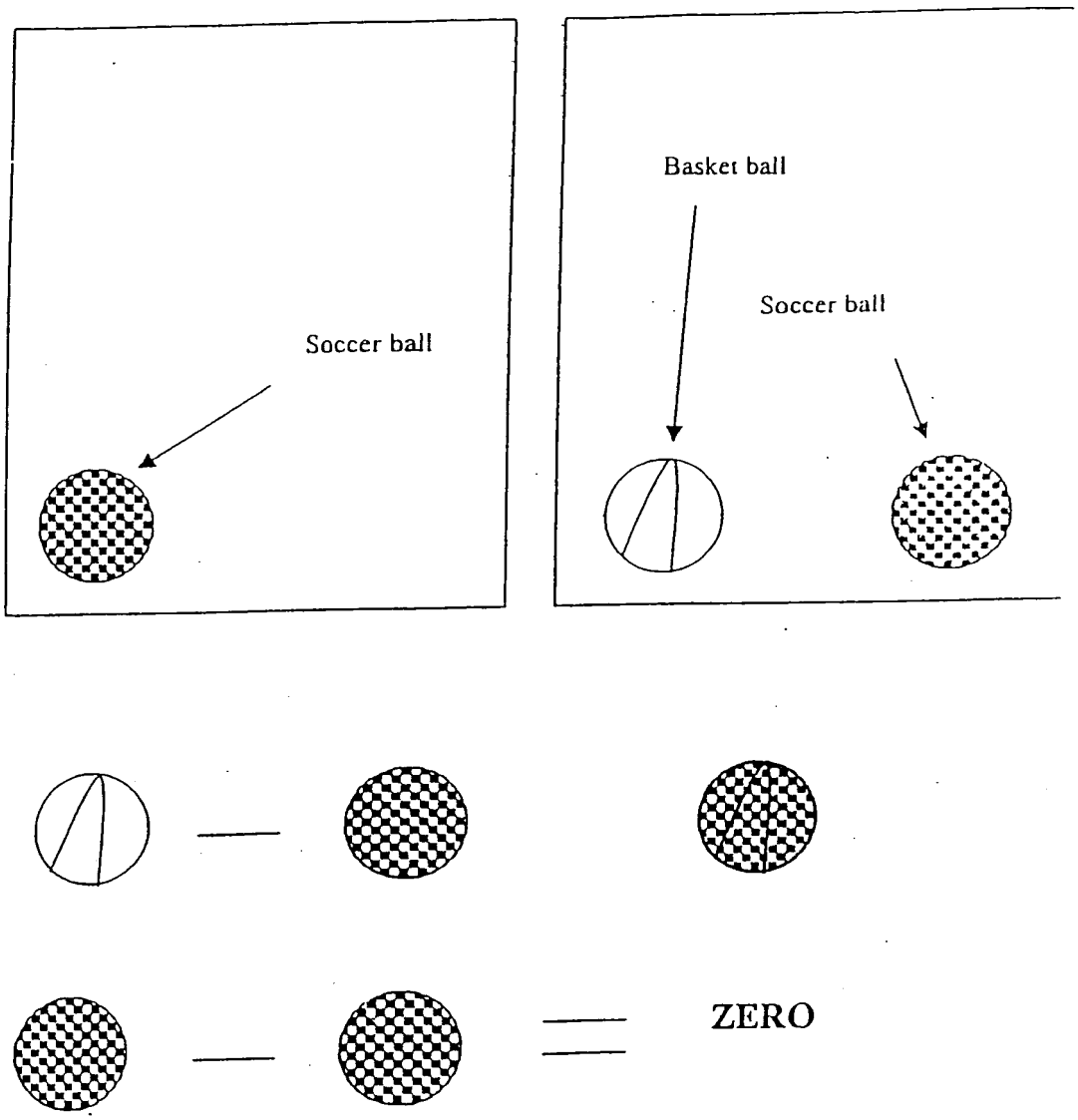

FIG. 7A

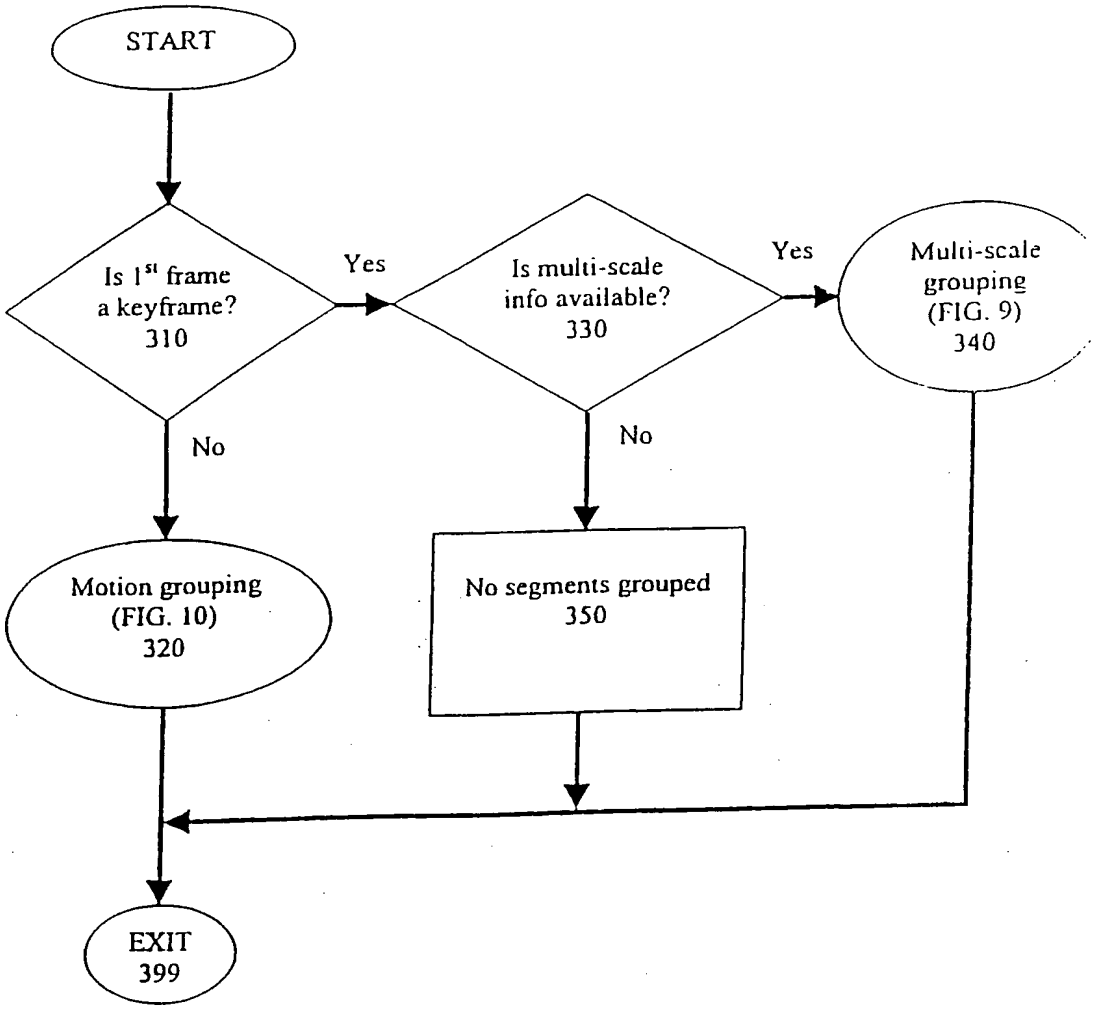
START
Is 1st frame a keyframe? 310
Yes
Is multi-scale info available? 330
Yes
Multi-scale grouping (FIG. 9) 340
No
No
Motion grouping (FIG. 10) 320
No segments grouped 350
EXIT 399

FIG. 8

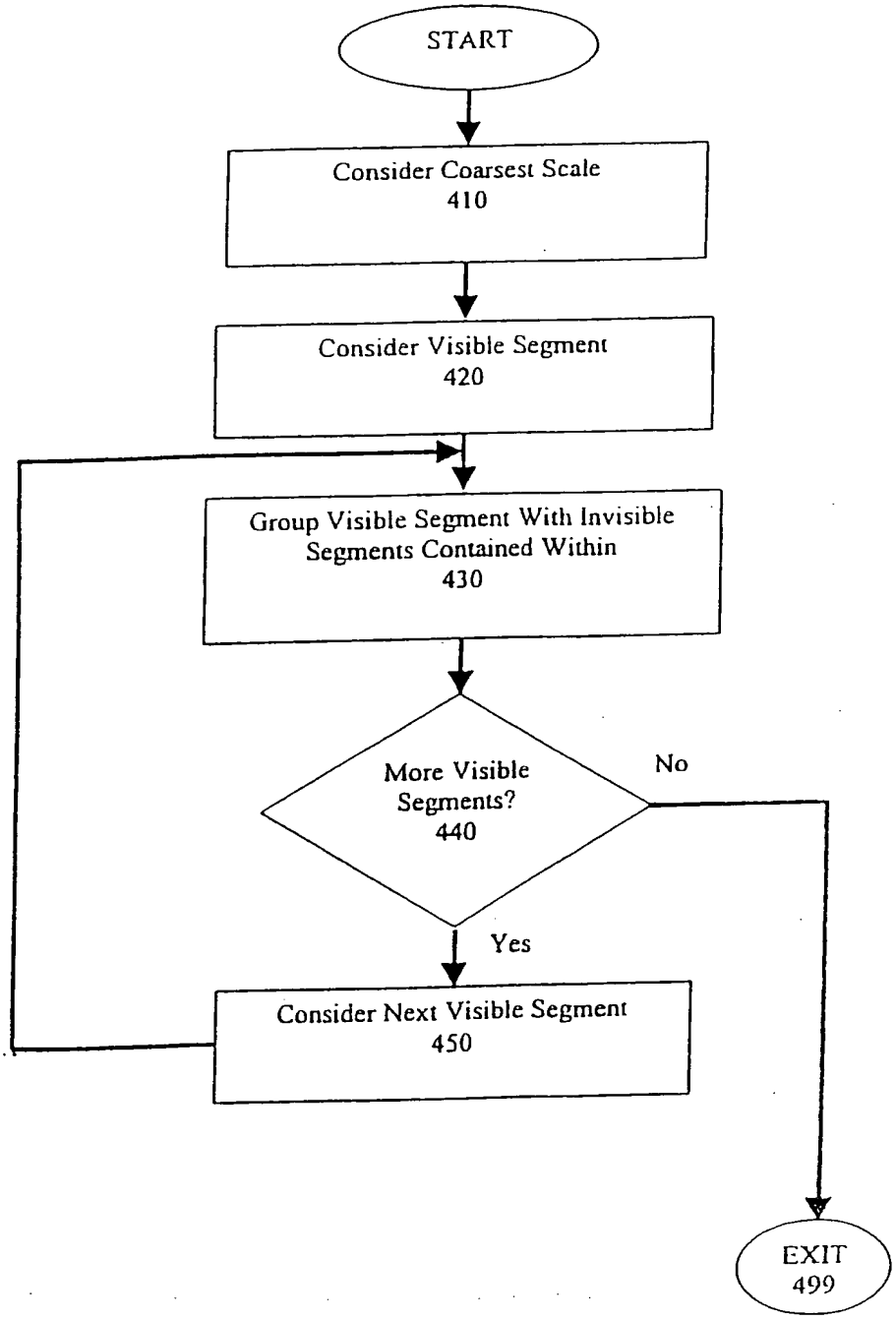
START
Consider Coarsest Scale
410
Consider Visible Segment
420
Group Visible Segment With Invisible
Segments Contained Within
430
More Visible
Segments?
440
No
Yes
Consider Next Visible Segment
450
EXIT
499

FIG. 9

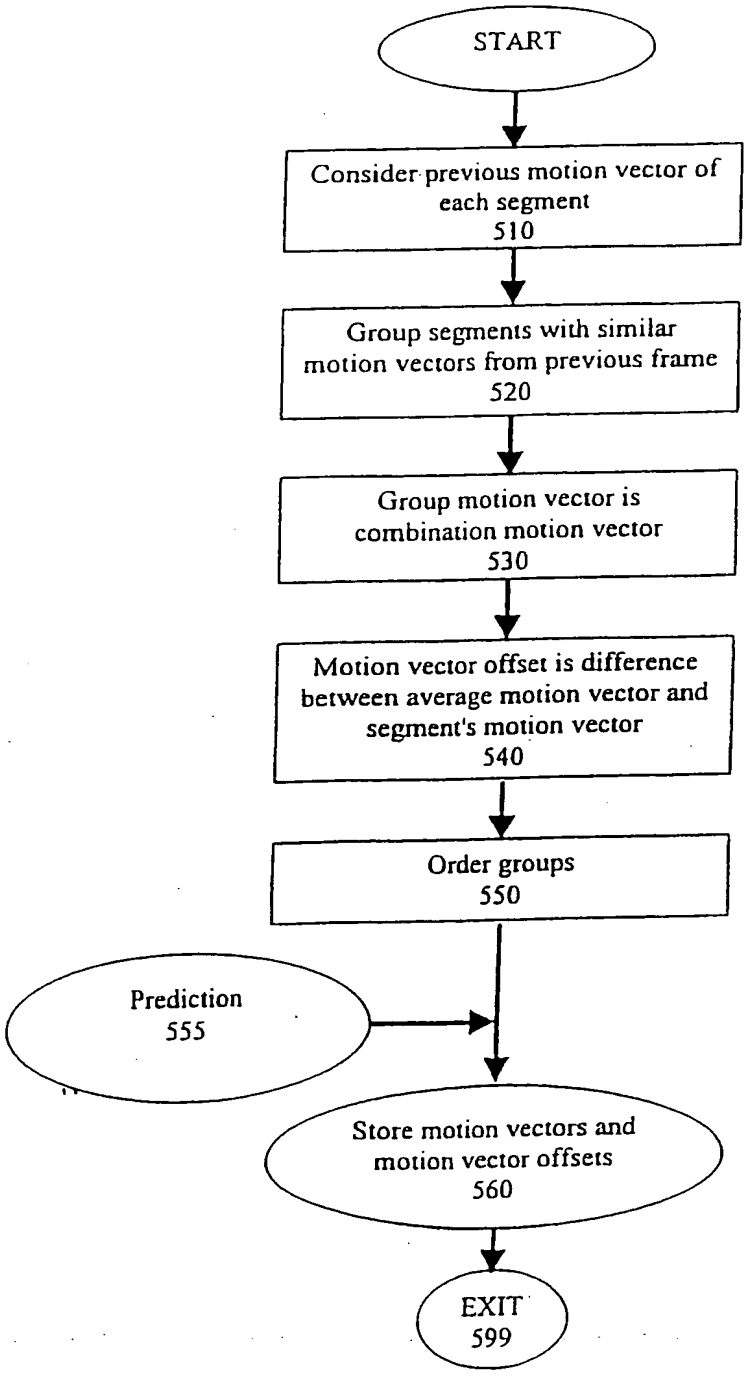
START
Consider previous motion vector of each segment
510
Group segments with similar motion vectors from previous frame
520
Group motion vector is combination motion vector
530
Motion vector offset is difference between average motion vector and segment's motion vector
540
Order groups
550
Prediction
555
Store motion vectors and motion vector offsets
560
EXIT
599

FIG. 10

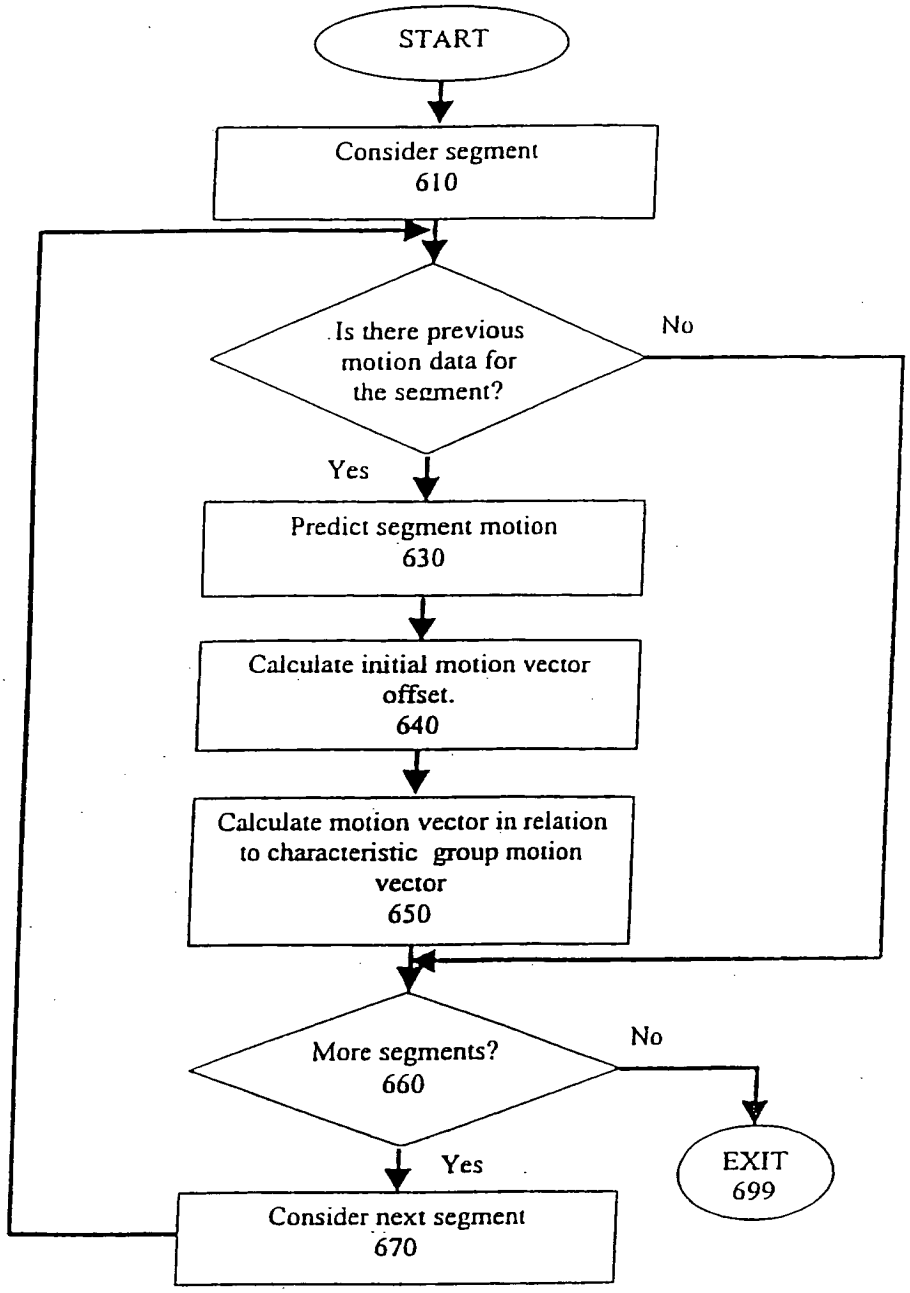
START
Consider segment
610
Is there previous motion data for the segment?
No
Yes
Predict segment motion
630
Calculate initial motion vector offset.
640
Calculate motion vector in relation to characteristic group motion vector
650
More segments?
660
No
Yes
EXIT
699
Consider next segment
670

FIG. 11

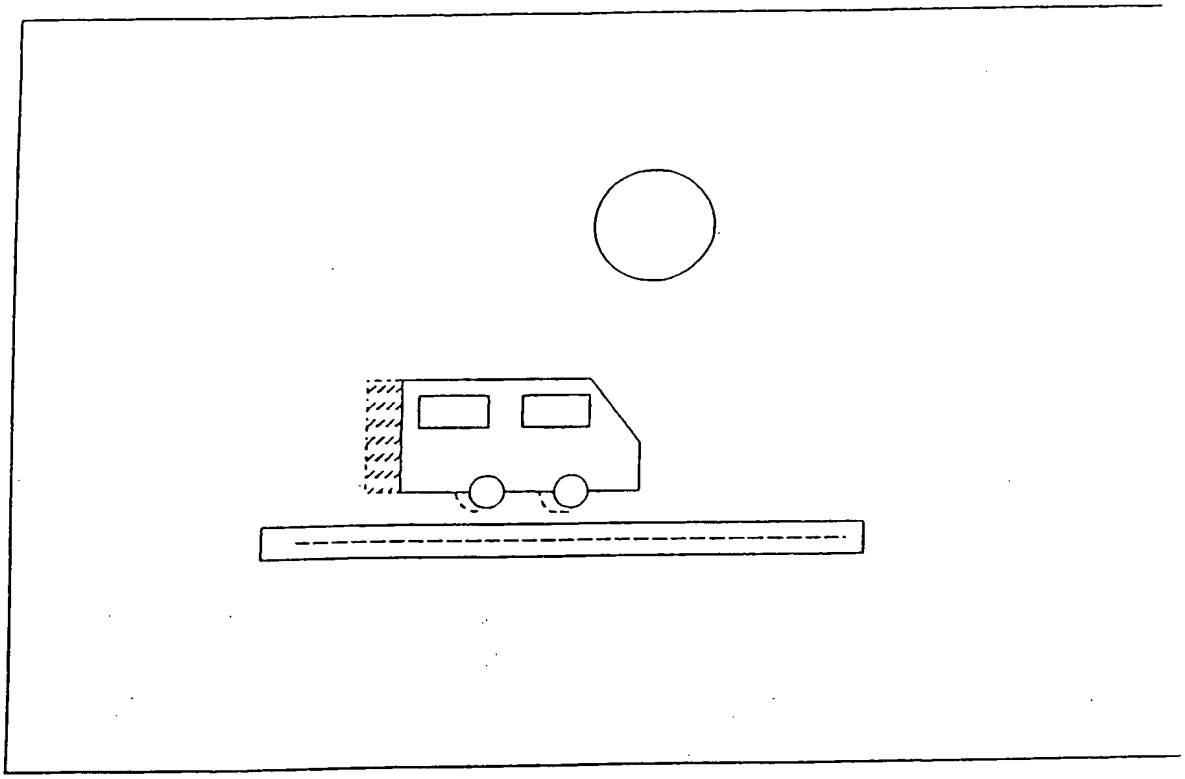

FIG. 12

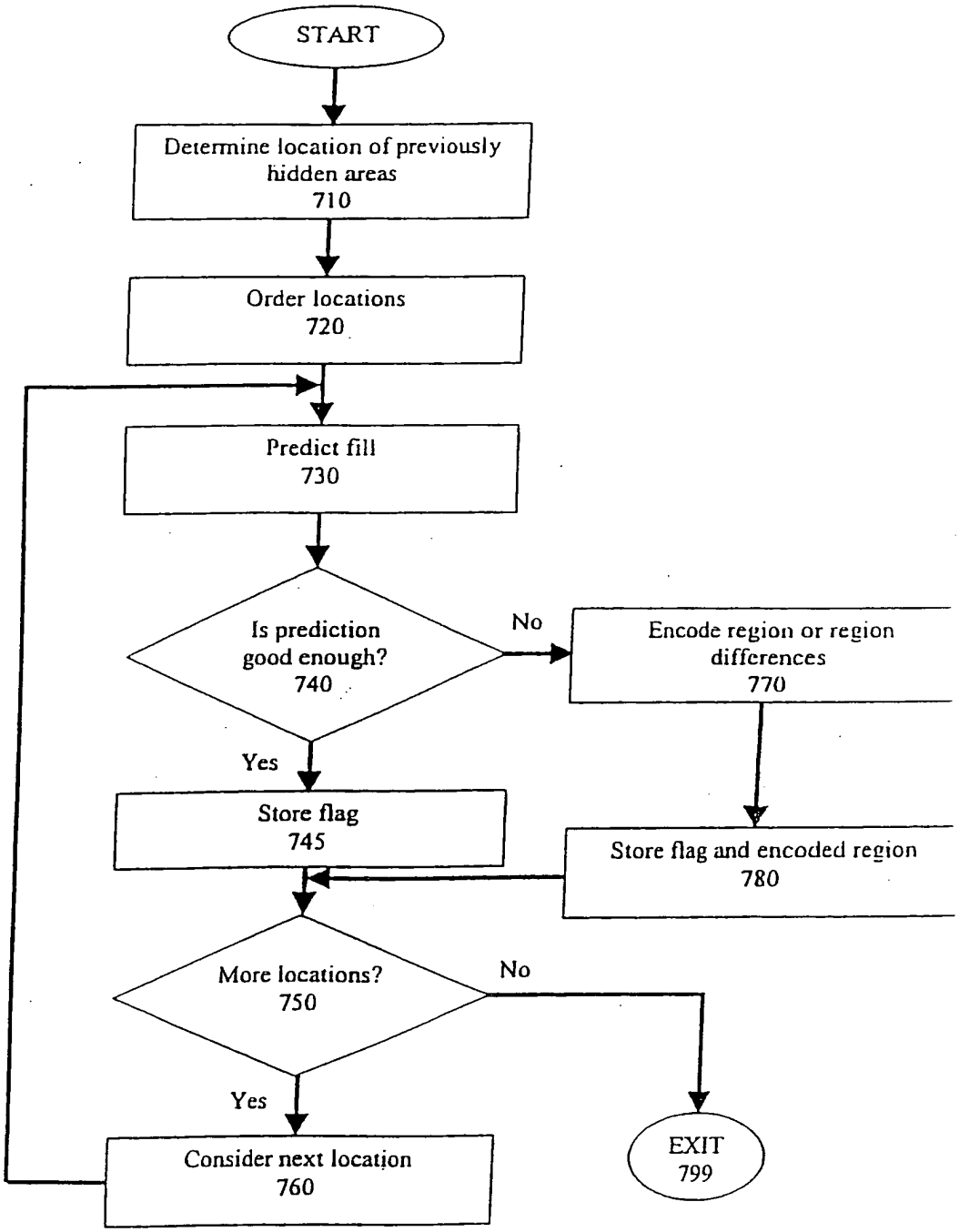
START
Determine location of previously hidden areas 710
Order locations 720
Predict fill 730
Is prediction good enough? 740
No
Encode region or region differences 770
Yes
Store flag 745
Store flag and encoded region 780
More locations? 750
No
Yes
Consider next location 760
EXIT 799

FIG. 13

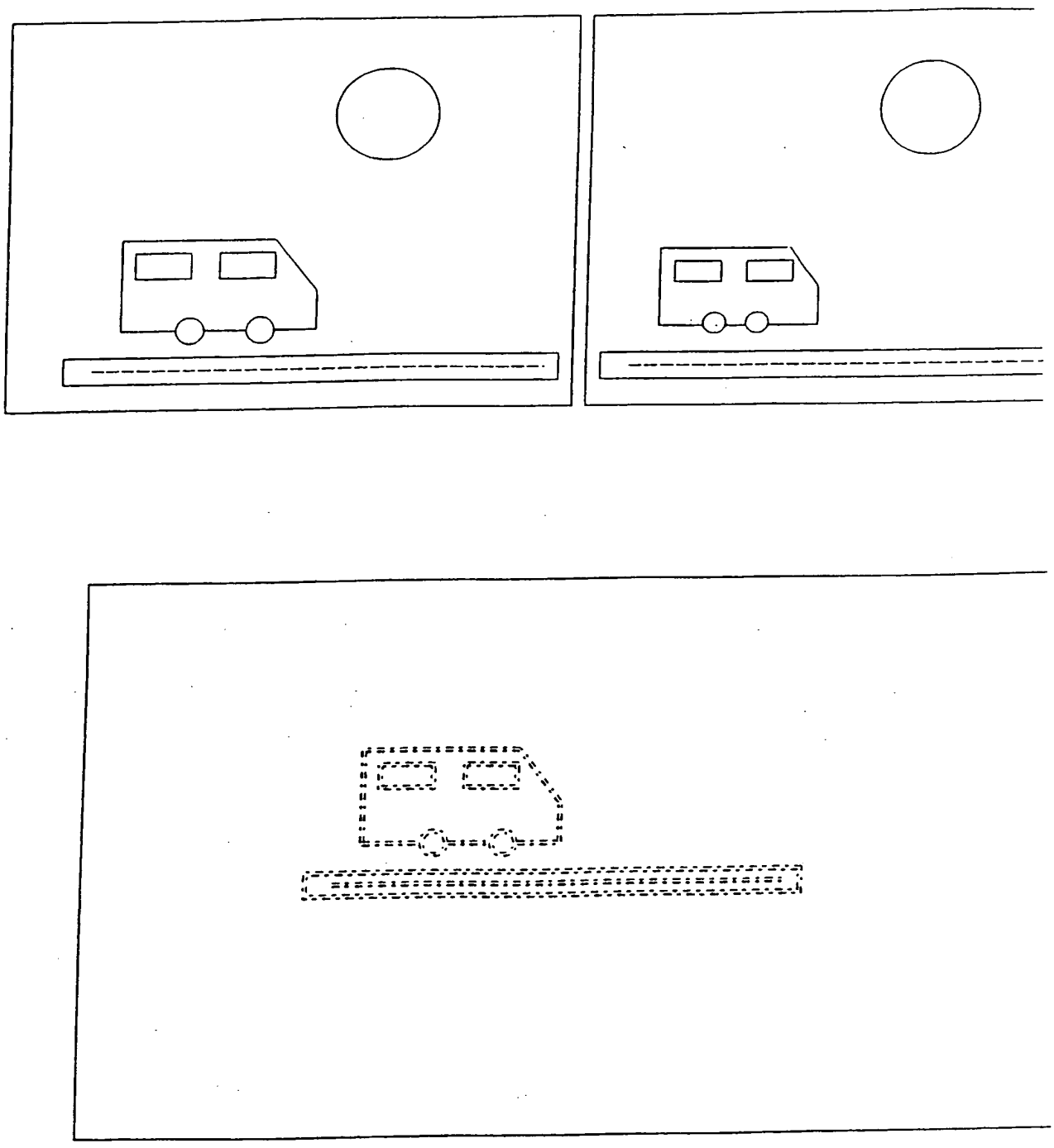

FIG. 14

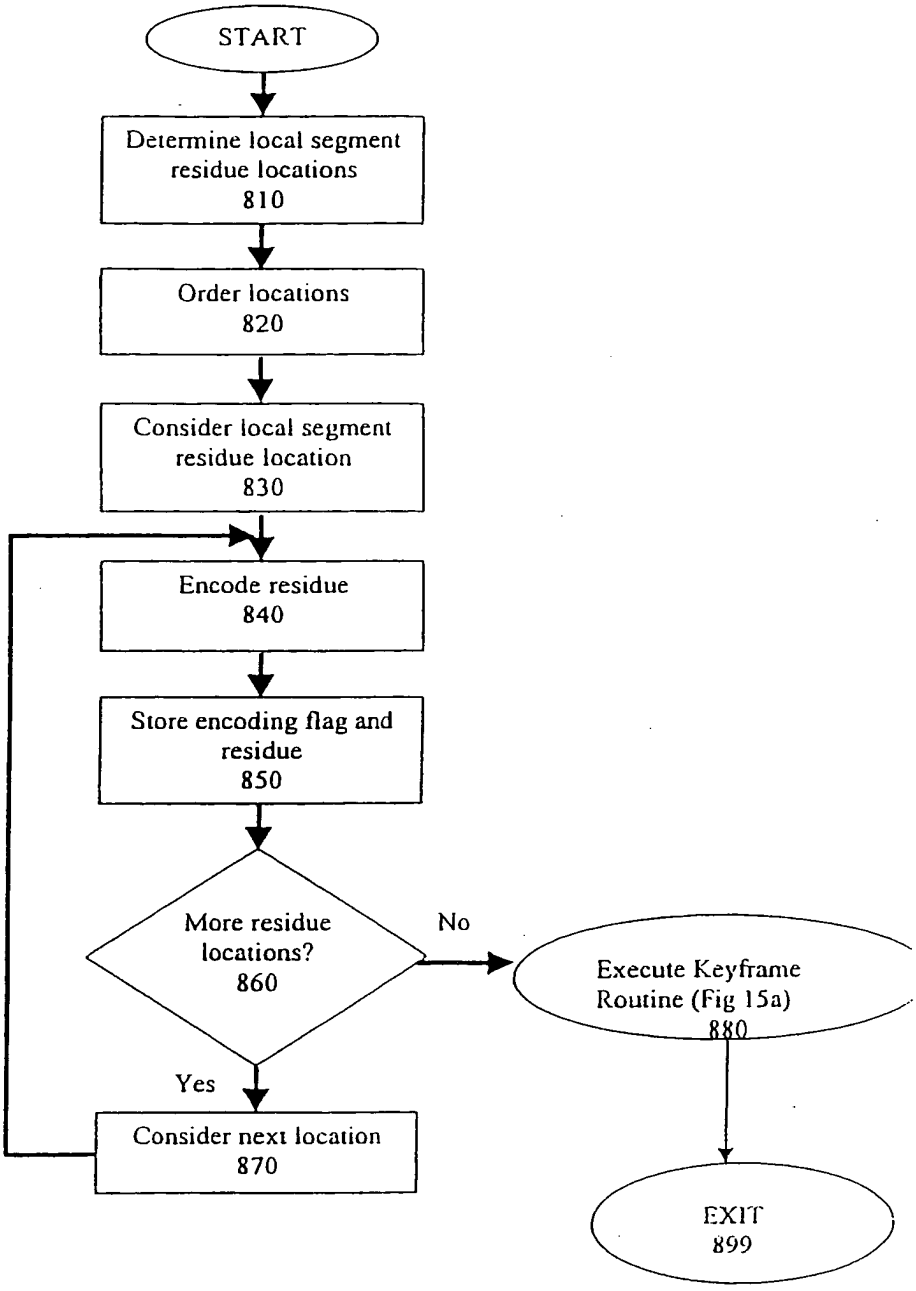
START
Determine local segment residue locations 810
Order locations 820
Consider local segment residue location 830
Encode residue 840
Store encoding flag and residue 850
More residue locations? 860
No
Yes
Execute Keyframe Routine (Fig 15a) 880
Consider next location 870
EXIT 899

FIG. 15

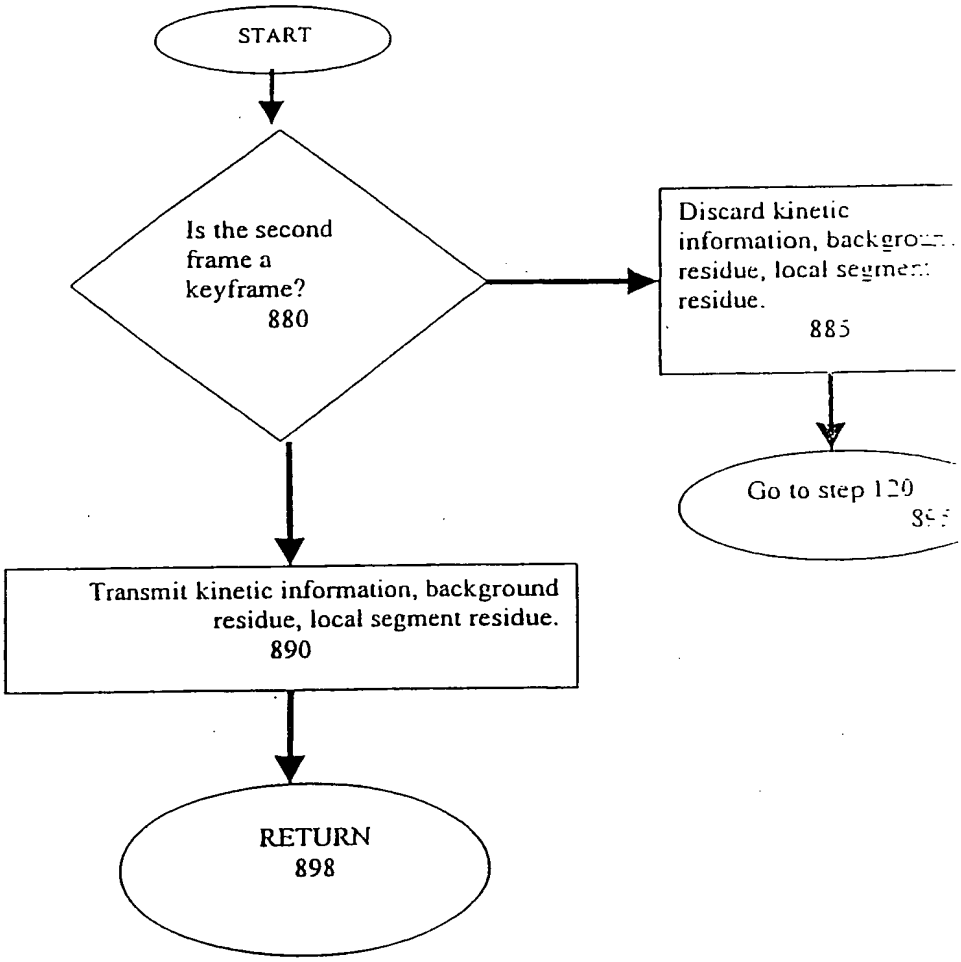
START
Is the second frame a keyframe?
880
Discard kinetic information, backgro... residue, local segme... residue.
885
Go to step 120
Transmit kinetic information, background residue, local segment residue.
890
RETURN
898

Fig. 15a

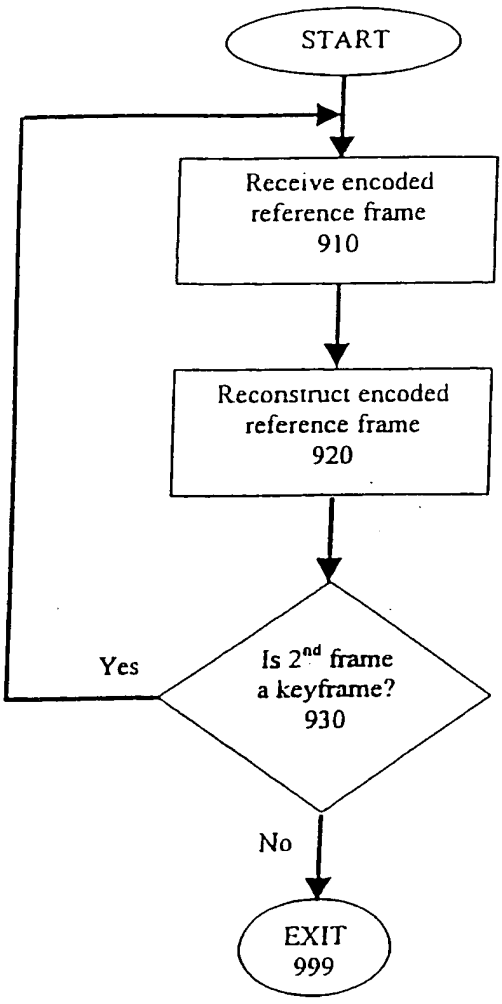
START
Receive encoded reference frame 910
Reconstruct encoded reference frame 920
Is 2nd frame a keyframe? 930
Yes
No
EXIT 999

FIG. 16

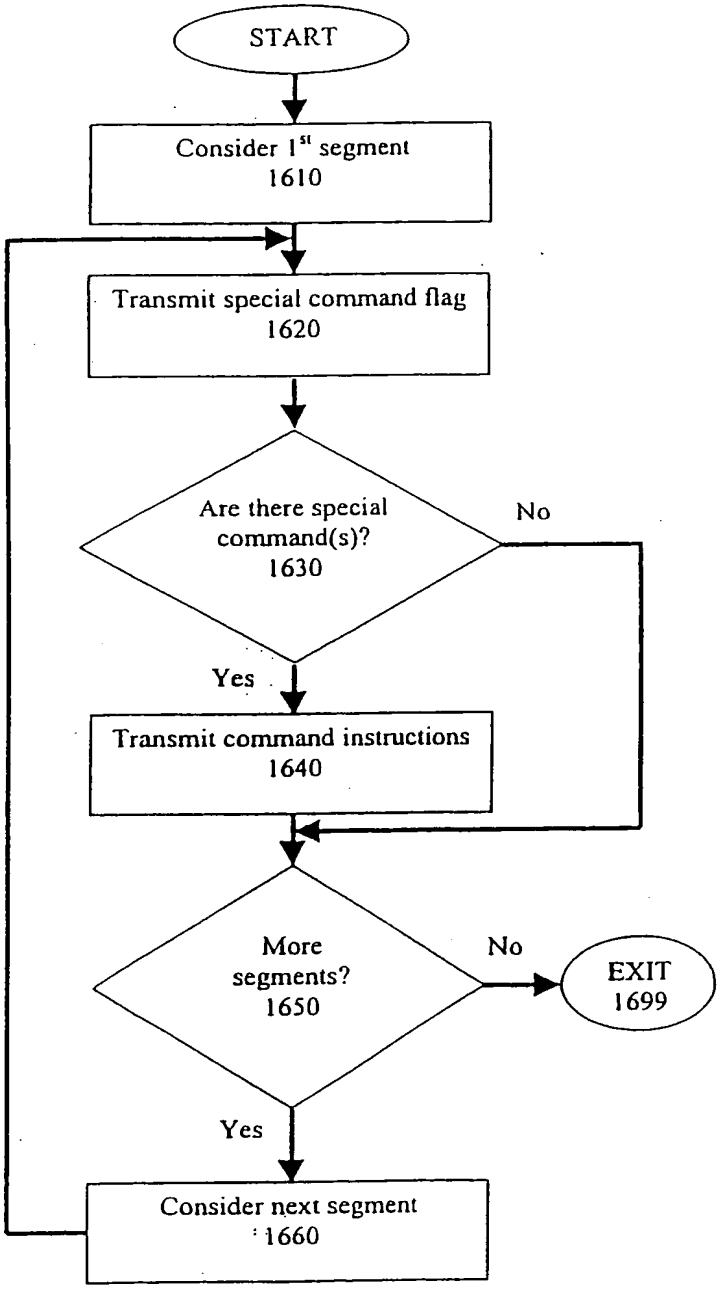
START
Consider 1st segment
1610
Transmit special command flag
1620
Are there special command(s)?
1630
No
Yes
Transmit command instructions
1640
More segments?
1650
No
EXIT
1699
Yes
Consider next segment
1660

FIG. 16.A

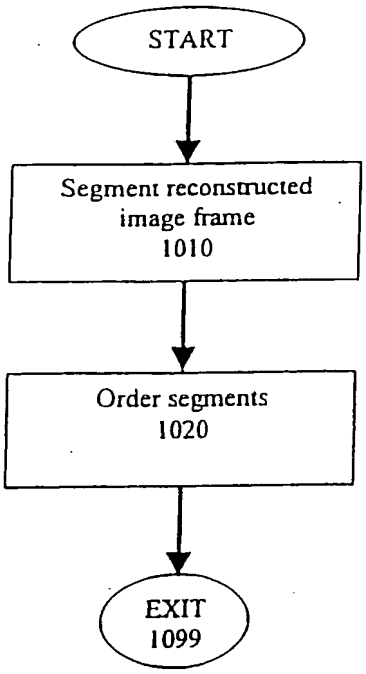
START
Segment reconstructed image frame 1010
Order segments 1020
EXIT 1099

FIG. 17

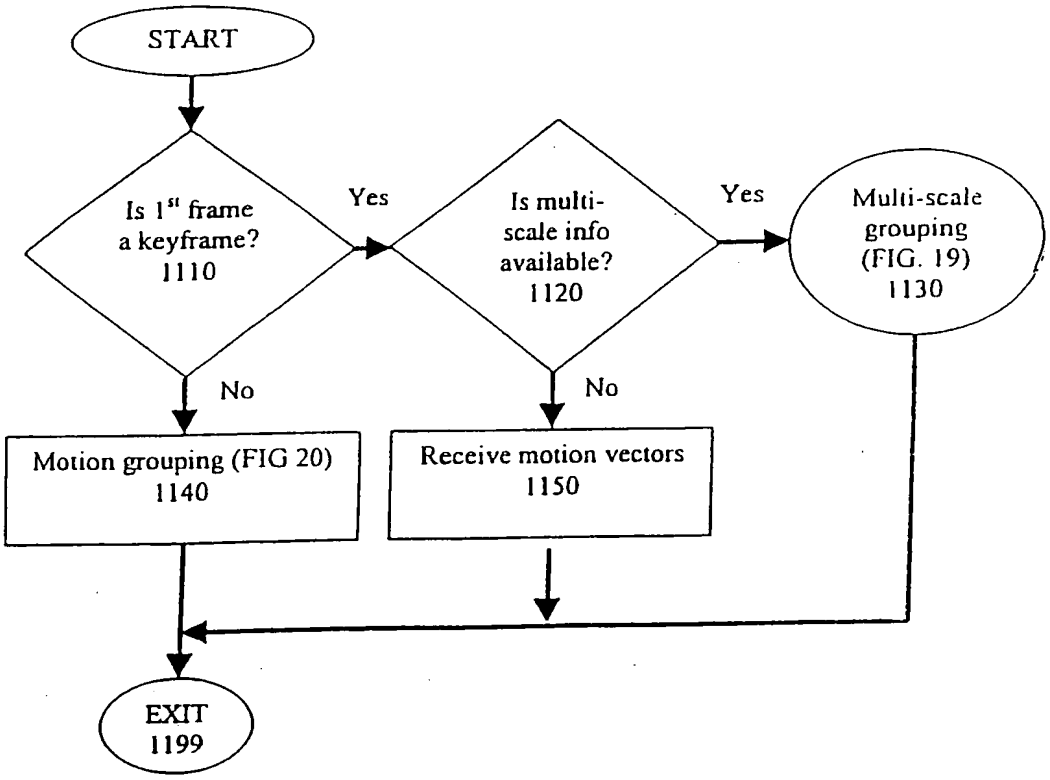
START
Is 1st frame a keyframe? 1110
Yes
Is multi-scale info available? 1120
Yes
Multi-scale grouping (FIG. 19) 1130
No
No
Motion grouping (FIG 20) 1140
Receive motion vectors 1150
EXIT 1199

FIG. 18

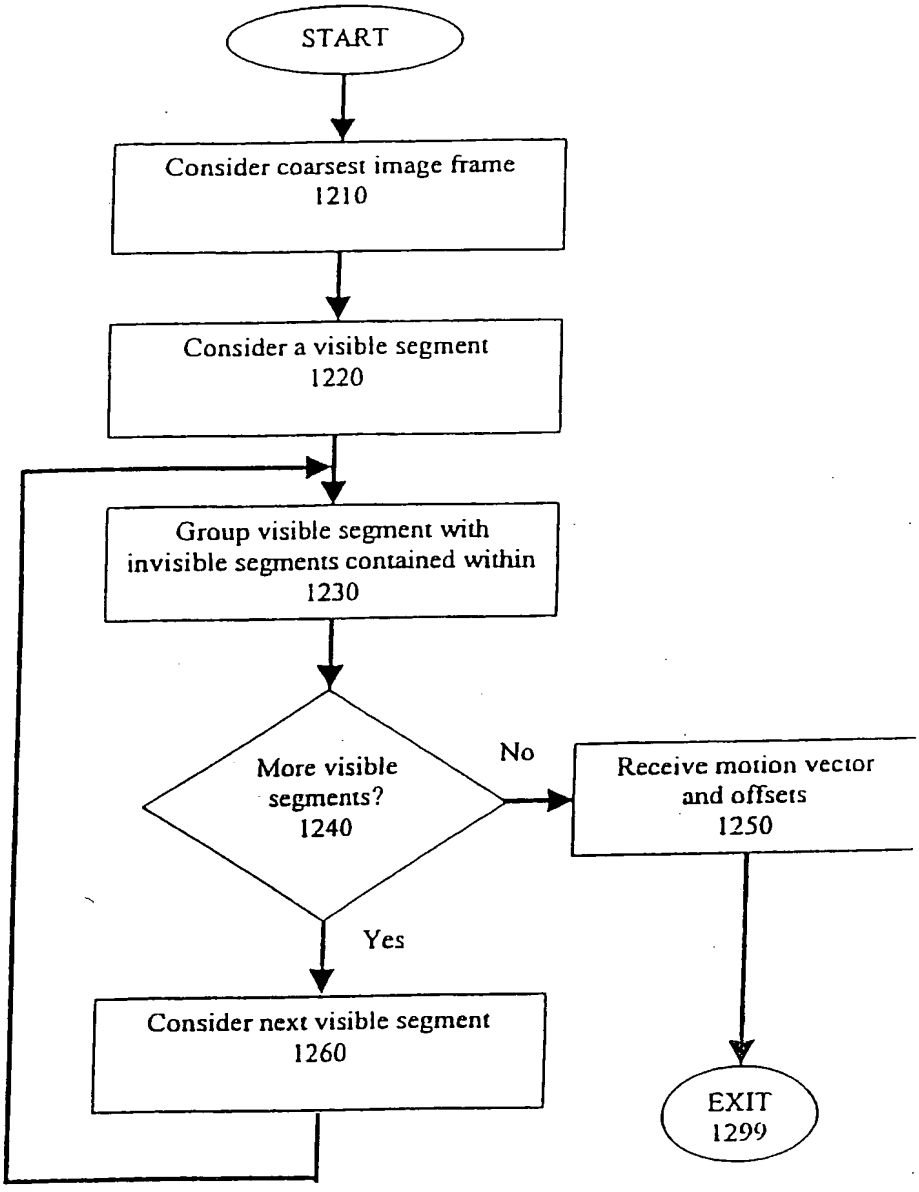
START
Consider coarsest image frame
1210
Consider a visible segment
1220
Group visible segment with
invisible segments contained within
1230
More visible
segments?
1240
No
Receive motion vector
and offsets
1250
Yes
Consider next visible segment
1260
EXIT
1299

FIG. 19

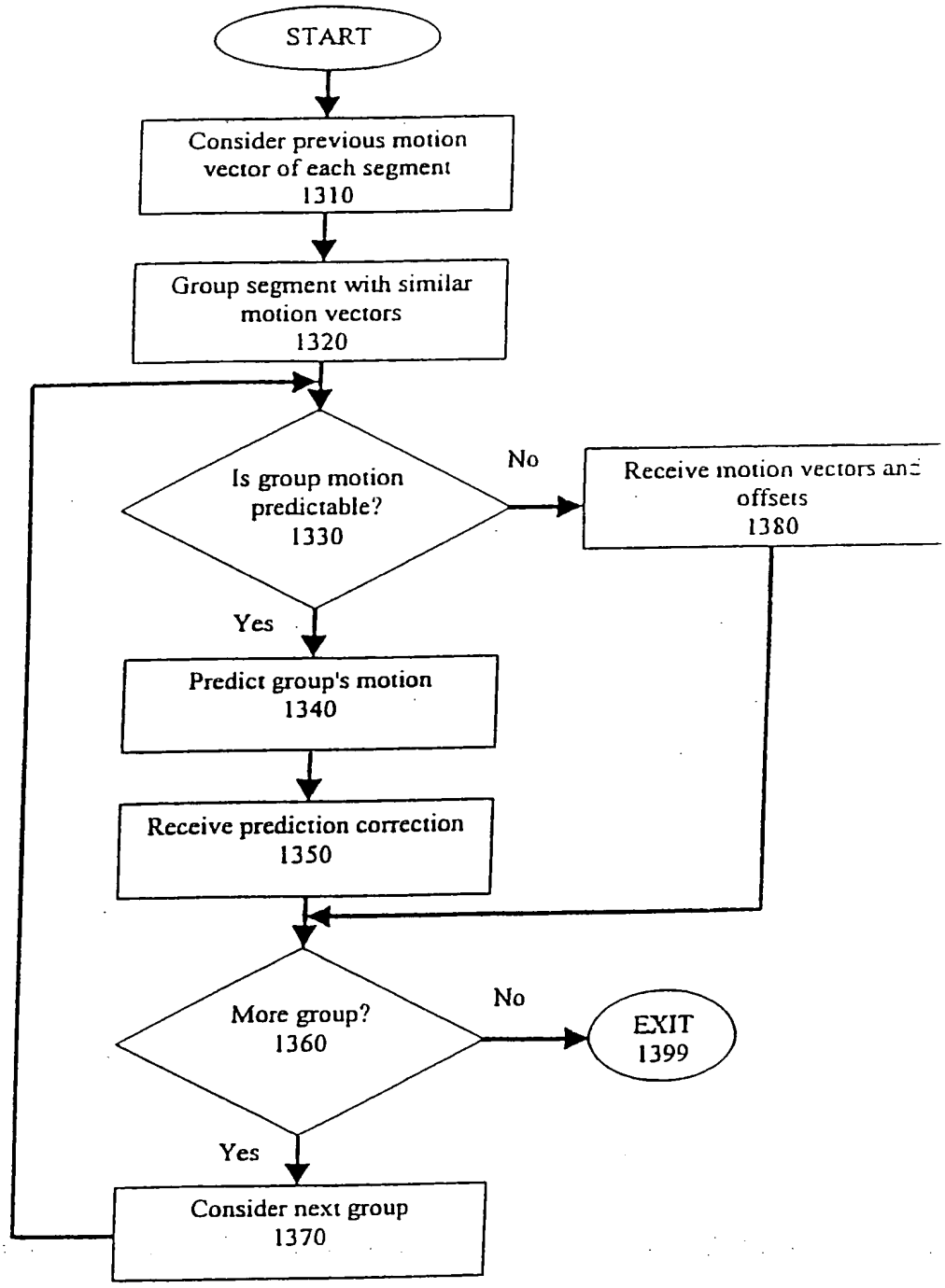
START
Consider previous motion vector of each segment 1310
Group segment with similar motion vectors 1320
Is group motion predictable? 1330
No
Receive motion vectors and offsets 1380
Yes
Predict group's motion 1340
Receive prediction correction 1350
More group? 1360
No
EXIT 1399
Yes
Consider next group 1370

FIG. 20A

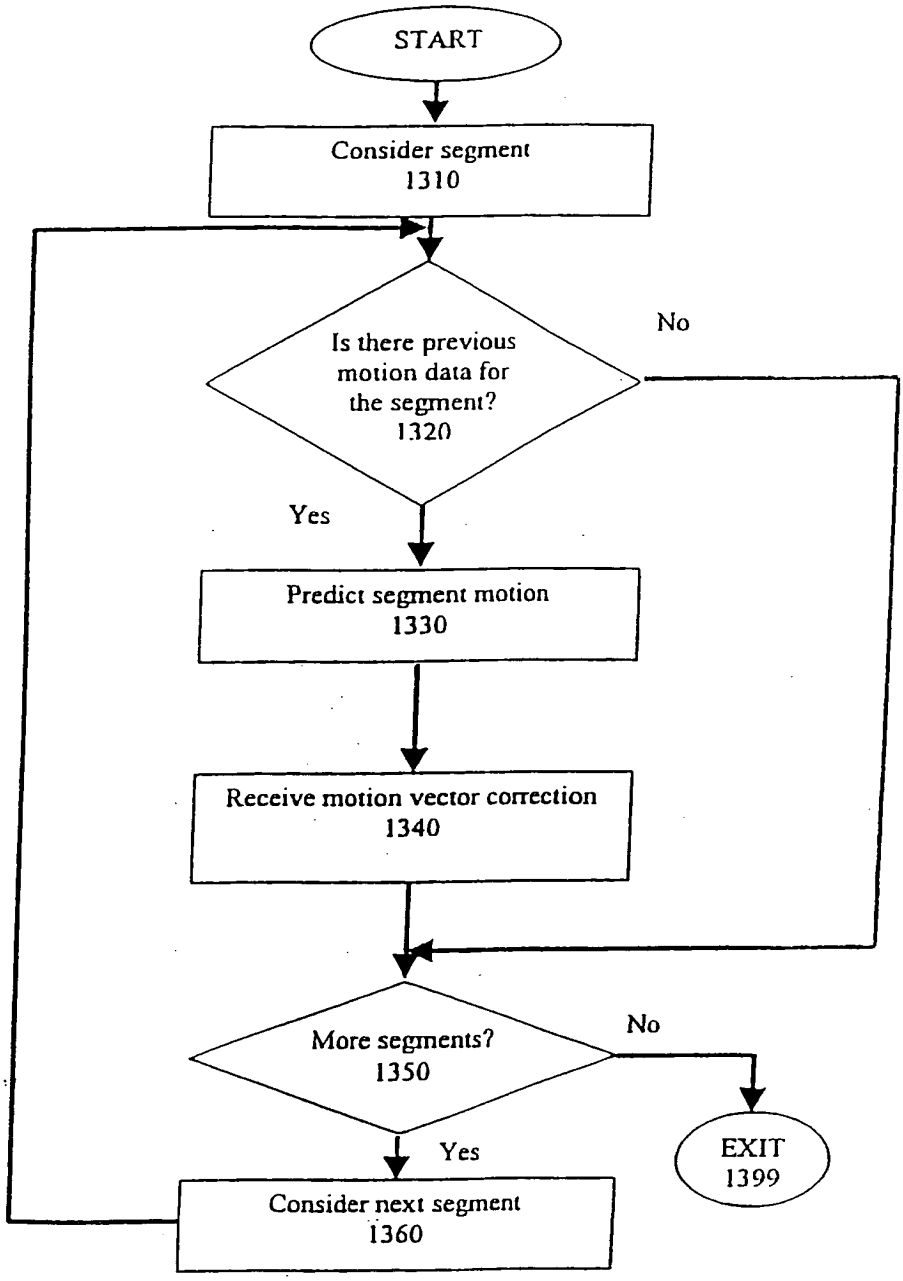
START
Consider segment
1310
Is there previous
motion data for
the segment?
1320
No
Yes
Predict segment motion
1330
Receive motion vector correction
1340
More segments?
1350
No
Yes
EXIT
1399
Consider next segment
1360

FIG. 20

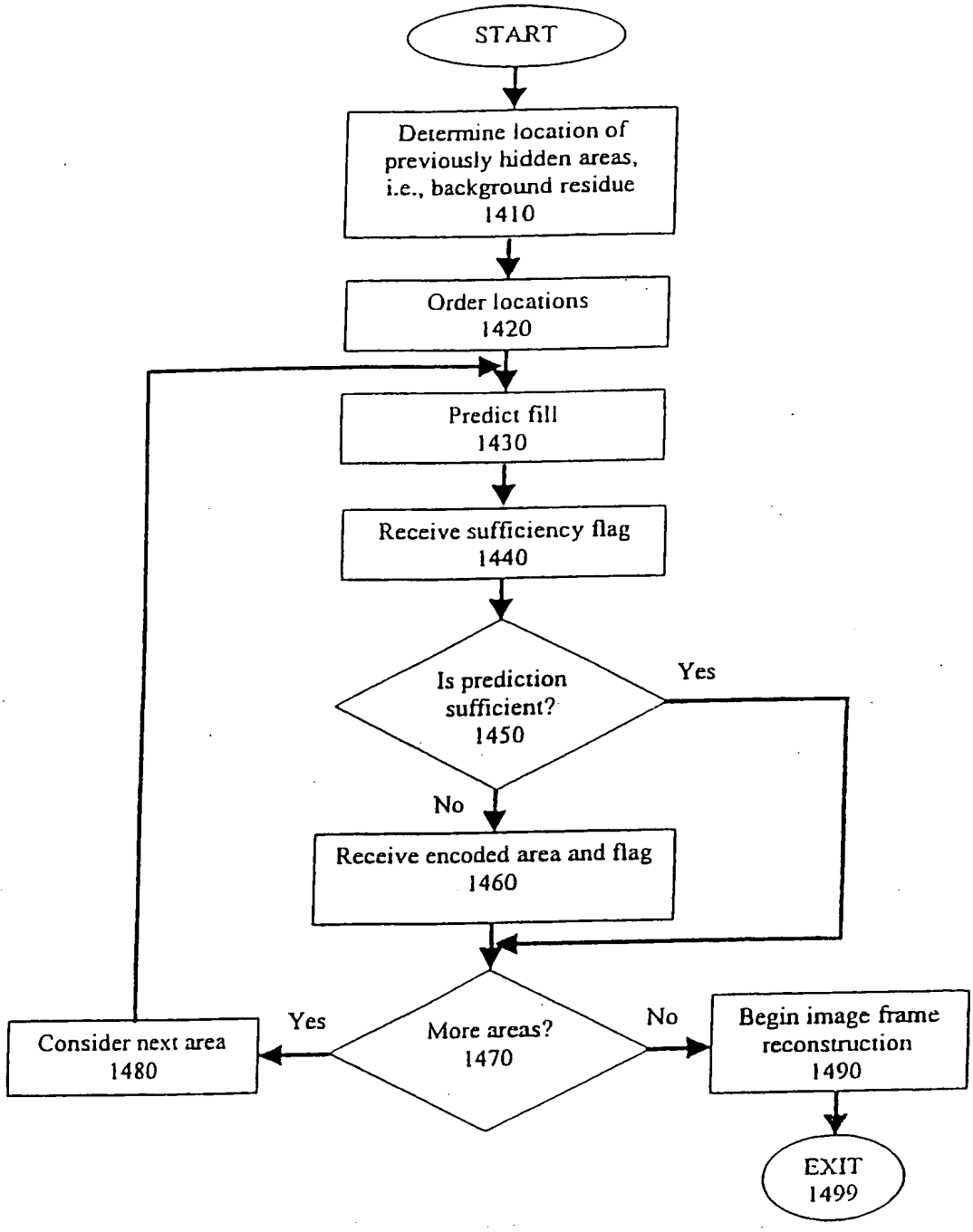
START
Determine location of previously hidden areas, i.e., background residue 1410
Order locations 1420
Predict fill 1430
Receive sufficiency flag 1440
Is prediction sufficient? 1450
Yes
No
Receive encoded area and flag 1460
More areas? 1470
Yes
No
Consider next area 1480
Begin image frame reconstruction 1490
EXIT 1499

FIG. 21

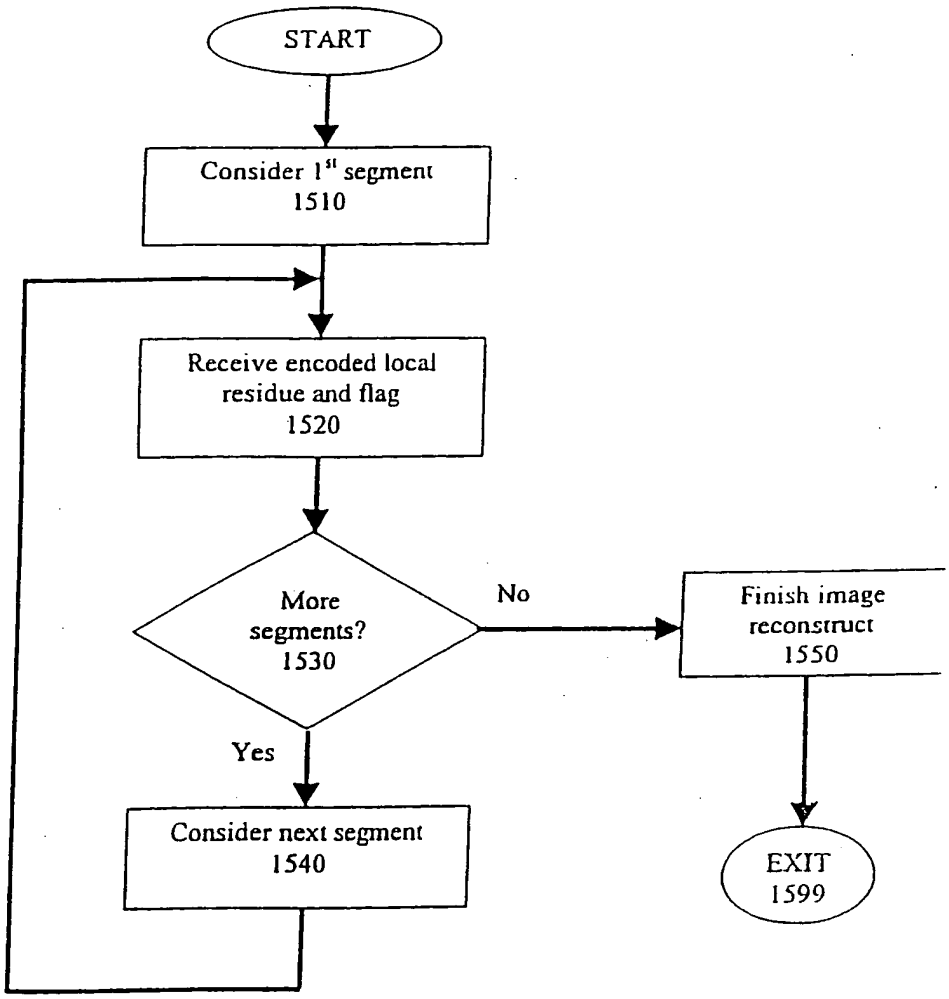
START
Consider 1st segment
1510
Receive encoded local residue and flag
1520
More segments?
1530
No
Finish image reconstruct
1550
Yes
Consider next segment
1540
EXIT
1599

FIG. 23

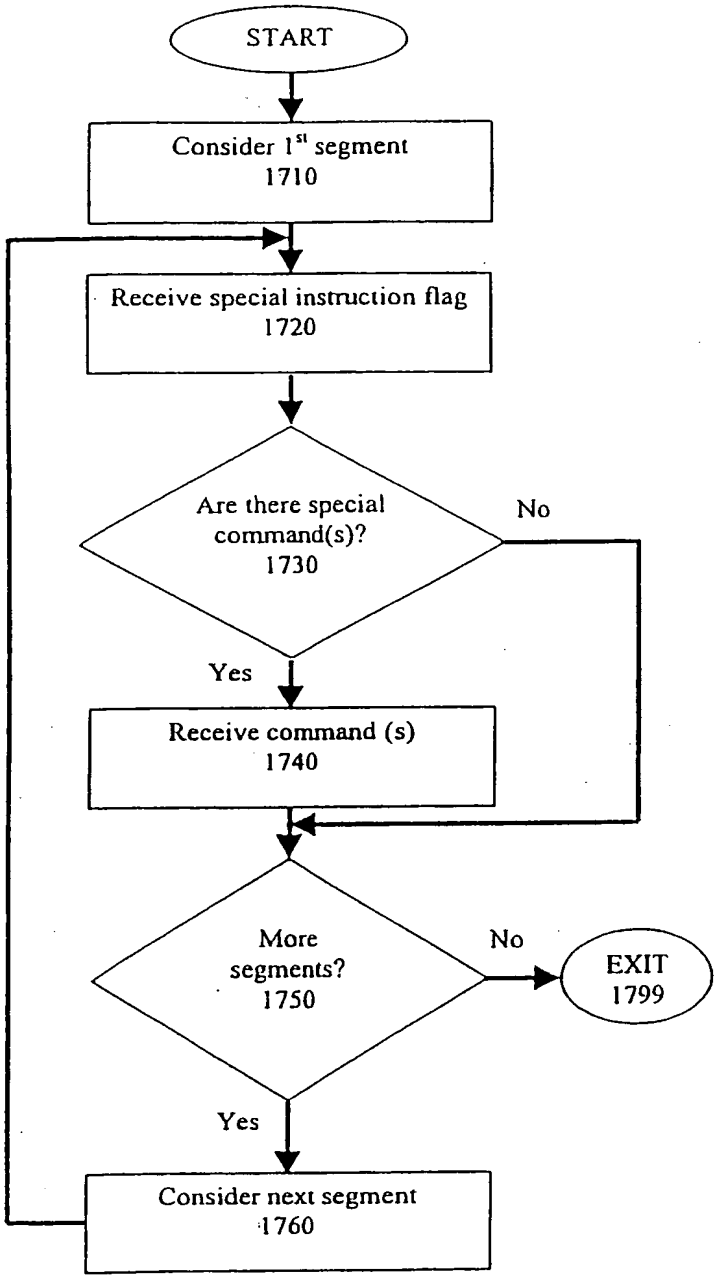
START
Consider 1st segment
1710
Receive special instruction flag
1720
Are there special command(s)?
1730
No
Yes
Receive command (s)
1740
More segments?
1750
No
EXIT
1799
Yes
Consider next segment
1760

FIG. 24

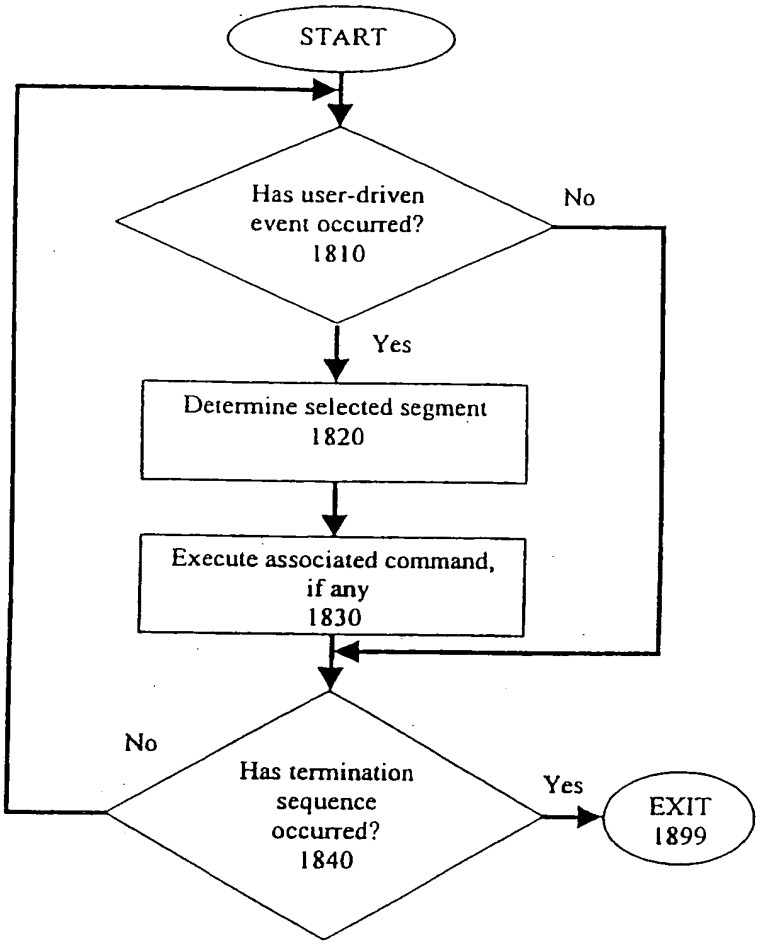
START
Has user-driven event occurred? 1810
No
Yes
Determine selected segment 1820
Execute associated command, if any 1830
No
Has termination sequence occurred? 1840
Yes
EXIT 1899

FIG. 25

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US00/10451

A. CLASSIFICATION OF SUBJECT MATTER

IPC(7) :H04N 5/262
US CL :375/240

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

U.S. : 375/240; 348/240

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X, P	US 6,026,182 A (LEE ET AL) 15 FEBRUARY 2000, col. 7, lines 43-52, col. 7, lines 63-67, col. 8, lines 1-5, col 10, lines 36-42, and Fig. 23b, element 706.	1.
X, E	US 6,057,884 A (CHEN ET AL) 02 MAY 2000, Fig. 3.	1.

☐ Further documents are listed in the continuation of Box C. ☐ See patent family annex.

*	Special categories of cited documents:	"T"	later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"A"	document defining the general state of the art which is not considered to be of particular relevance	"X"	document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"E"	earlier document published on or after the international filing date	"Y"	document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
"L"	document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)	"&"	document member of the same patent family
"O"	document referring to an oral disclosure, use, exhibition or other means		
"P"	document published prior to the international filing date but later than the priority date claimed		

Date of the actual completion of the international search 22 JUNE 2000	Date of mailing of the international search report 26 JUL 2000
Name and mailing address of the ISA/US Commissioner of Patents and Trademarks Box PCT Washington, D.C. 20231 Facsimile No. (703) 305-3230	Authorized officer CHRISTOPHER KELLEY James R. Matthews Telephone No. (703) 305-4856

Form PCT/ISA/210 (second sheet) (July 1998)*

(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

CORRECTED VERSION

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
26 October 2000 (26.10.2000)

PCT

(10) International Publication Number
WO 00/64148 A1

(51) International Patent Classification[7]: H04N 5/262

(21) International Application Number: PCT/US00/10451

(22) International Filing Date: 17 April 2000 (17.04.2000)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
60/129,854 17 April 1999 (17.04.1999) US

(71) Applicant *(for all designated States except US)*: **PULSENT CORPORATION** [US/US]: 1455 McCarthy Boulevard, Milpitas, CA 95035 (US).

(72) Inventors; and

(75) Inventors/Applicants *(for US only)*: **PRAKASH, Adityo** [IN/US]: 600 Marlin Court, Redwood Shores, CA 94065-1267 (US). **PRAKASH, Eniko, F.** [RO/US]: 600 Marlin Court, Redwood Shores, CA 94065-1267 (US).

(74) Agents: **ALBERT, Philip, H.** et al.; Townsend and Townsend and Crew LLP, 8th floor, Two Embarcadero Center, San Francisco, CA 94111 (US).

(81) Designated States *(national)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, CA, CH, CN, CR, CU, CZ, DE, DK, DM, DZ, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, TZ, UA, UG, US, UZ, VN, YU, ZA, ZW.

[Continued on next page]

(54) Title: METHOD AND APPARATUS FOR EFFICIENT VIDEO PROCESSING

ENCODER/DECODER SYSTEM

	Encoder		Decoder
1	Obtain encode, transmit frame	→	Receive Frame
2	Reconstruct Frame		Reconstruct Frame
3	Segmentation		Segmentation
4	Order segments		Order Segments
5	Obtain new image frame		
6	Determine segment motion		
7	Encode motion information		
8	Determine background residue		
9	Predict background residue fill		
10	Determine sufficiency of prediction		
11	Determine local residue		
12	Order local residue locations		
13	Encode residue		
14	Is 2nd frame keyframe, yes, goto 5	→	Receive Keyframe Flag
15	Transmit motion data	→	Receive motion data
16			Determine and order background residue
17			Predict background residue
18	Transmit background residue data	→	Receive additional background residue data
19			Determine and order local segment residues
20	Transmit local segment residue	→	Receive local segment residue
21	Reconstruct 2nd frame		Reconstruct 2nd frame
22	Goto Step 5		Goto Step 5

(57) Abstract: A video compression method and apparatus is disclosed. The present invention includes a "smart" or active decoder (Fig. 3) that performs much of the transmission and the instruction burden that would otherwise be required of the encoder, thus greatly reducing the overhead and resulting in a much smaller encoded bitstream. Thus, the corresponding (i.e., compatible) encoder of the present invention can produce an encoded bitstream with a greatly reduced overhead. This is achieved by encoding a reference frame (Fig. 3, element 7) based on the structural information inherent to the image (e.g., image segmentation, geometry, color, and/or brightness), and then predicting other frames relative to the structural information. Typically, the description of a predicted frame would include kinetic information (Fig. 3, element 6) (e.g., segment motion data and/or inexact matches and appearance of new information, and portion of the segment evolution that is captured by motion per se etc.). Because the decoder is capable of independently determining the structural information (and relationships thereamong) underlying the predicted frame, such information need not be explicitly transmitted to the decoder. Rather, the encoder need only send information that the encoder knows the decoder cannot determine on its own.

WO 00/64148 A1

(84) **Designated States** *(regional)*: ARIPO patent (GH, GM, KE, LS, MW, SD, SL, SZ, TZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).

Published:
— *with international search report*

(48) **Date of publication of this corrected version:**
2 May 2002

(15) **Information about Correction:**
see PCT Gazette No. 18/2002 of 2 May 2002, Section II

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

METHOD AND APPARATUS FOR EFFICIENT VIDEO PROCESSING

1. Brief Introduction

The present invention relates to the compression of motion video data, and more particularly for a synchronized encoder and smart decoder system for the efficient transmittal and storage of motion video data. As consumers desire more motion video intensive modes of communications, the limited bandwidth of current transmission modes, such as broadcast, cable, telephone lines, etc. becomes prohibitive. The introductions of the Internet, and the subsequent popularity the world wide web, video conferencing, digital and interactive television require more efficient ways of utilizing existing bandwidth. Further, motion video intensive applications require immense storage capacity. The advent of multi-media capabilities on most computer systems have taxed tradition storage devices such as hard drives, to the limit.

Compression, as used in this patent, is the means by which digital motion video can be represented efficiently and cheaply. The ultimate goal of video compression is to reduce the bitstream, or video information flow, of the motion video sequences as much as possible, while retaining enough information so that the decoder or receiver can reconstruct the video image sequences in a manner adequate for the specific application, such as television, videoconferencing, etc. The benefit of compression is that it allows more information to be transmitted in a given amount of time, or stored in a given storage medium.

Most digital signals contain a substantial amount of redundant, superfluous, information. For example, a stationary video scene produces nearly identical images in each scene. Compression attempts to remove the superfluous information so that the

related image frames can be represented in terms of the previous, thus eliminating the need to transmit the entire scene for each video frame.

2. Previous attempts

There have been numerous attempts at adequately compressing video imagery. These methods generally fall into one of the following two categories: 1) Spatial redundancy reduction, and 2) Temporal redundancy reduction.

2.1 Spatial Redundancy Removal

The first type of video compression focuses on the reduction of spatial redundancy. Spatial redundancy refers to taking advantage of the correlation among neighboring pixels in order to derive a more efficient representation of the important information in an image frame. These methods are more appropriately termed still image compression routines, as they do not attempt to address the issue of temporal, or frame to frame, redundancy, as explained in section 2.2. They work reasonably well on individual video image frames. However, a critical element in video compression is reducing temporal redundancy, in other words, not having to retransmit, store, or otherwise fully represent, information seen in previous frames. Common still image compression schemes include JPEC, Wavelets, and Fractals.

2.1.1 JPEG/DCT based image compression

One of the first commonly used methods of image compression was the DCT, or direct cosine transformation, compression system, which is at the heart of JPEG.

DCT operates by representing each digital image frame as a series of cosine waves or frequencies. Afterwards, the coefficients of the cosine series are quantized. The higher frequency coefficients are quantized more harshly than those of the lower frequencies are. The result of the quantization is large number of zero coefficients, which

can be encoded very efficiently. However, JPEG and similar compression schemes do not address this crucial issue of temporal redundancy.

2.1.2 Wavelets

As a slight improvement to the DCT compression scheme, the wavelet transformation compression scheme was devised. This system is similar to the DCT. The only substantial difference is that the image frame is represented as a series of wavelets, or windowed oscillations, instead of as a series of cosine waves.

2.1.3 Fractals

The goal of fractal compression is to take an image and determine the single function or set of functions, which fully describe the image frame. A fractal is an object that is self-similar at different scales, or resolutions, i.e. no matter what resolution you look at, the object remains the same. Theoretically, fantastic compression ratios could occur as simple equations describe complex images.

Fractal compression is not a viable method of general compression. The high compression rations only work on specially constructed images, and only with considerable help from a person guiding the compression process. Fractal Compression is a computationally intensive process.

2.2 Temporal and Spatial Redundancy Removal

Adequate motion video compression requires reduction of both temporal and spatial redundancies within the sequence of frames that comprise video. Temporal redundancy removal is concerned with the removal from the bitstream, information that had already been coded in previous image frames. Block matching is the basis for most currently used effective means of temporal redundancy removal.

2.2.1 Block Based Motion Estimation

Block Matching is the process by which a block of the image is subdivided into uniform size blocks and each block is tracked from one frame to another and represented by a motion vector instead of having the block re-coded and placed into the bitstream for a second time. Examples of compression routines that use block matching include MPEG, and all its variants.

MPEG operates by performing a still image compression on the first frame and transmitting it. It then divides the same frame into 16 pixel by 16 pixel square blocks and attempts to find each block within the next frame. For each block that still exists in the subsequent frame, MPEG needs only transmit the motion vector, or movement, of the block along with sufficient identifying information. As the block moves from frame to frame, it may not remain the same. The difference is known as the residue. Additionally, as blocks move, previously hidden areas may become visible for the first time. This is also known as the residue. Collectively, the remaining information after the block motion is sent is known as the residue frame, which is coded using JPEG and sent to the receiver to complete the image frame.

Next, the encoder divides the second image frame into blocks and the routine continues until a new keyframe is inserted. A keyframe is an image frame which is completely self-contained, not described in relation to any other image frame.

Although state of the art, block matching is highly inefficient and fails to take advantage of the known general physical characteristics of images. For example, the block method is inherently crude, as the blocks do not have any relationship with real objects in the image. A given block may comprise a part of an object, a whole object, or even multiple dissimilar objects with unrelated motion. In addition, often, neighboring

objects will have similar motion. However, since blocks do not correspond to real objects, block based systems cannot use this information to further reduce the bitstream

Another major limitation of block based matches is the residue frame coding. The residue frame created after block based matching will generally be noisy and patchy and does not lend itself to good compression via standard image compression schemes such as DCT, wavelets, or fractals.

2.3 Alternatives

It is well recognized that the current state of the art needs improvement, specifically the block based method is extremely inefficient and does not produce an optimally compressed bitstream for motion video information. To that end, the latest compression schemes, such as MPEG4 allows for the inclusion of the structural information, if available, of selected items within the frames instead of merely using arbitrary sized blocks. While, some compression gains are achieved, the overhead information is substantially increased because in addition to the motion and residue information these schemes require that the structural or shape information for each item must be sent to the receiver. This is because all current compression schemes use a dumb receiver, one, which is incapable of making determinations for itself.

Additionally, as mentioned above, the current compression methods code the residue frame merely another image frame to be compressed by JPEG, without attempting to determine is more efficient methods are possible.

3. Novel Approaches

This invention represents a novel approach to the problem of video compression. As described above, the goal of video compression is to represent accurately a sequence of video frames with the smallest bitstream, or video information flow. As previously

stated, spatial redundancy reduction methods above are inappropriate for motion video compression. Further, the current temporal and spatial redundancy reduction methods such as MPEG2 waste precious bitstream space by having to transmit a lot of overhead information. This invention solves that problem by using a smart decoder. This smart decoder determines much of the overhead information, thus obviating the necessity of transmitting such information, and therefore reducing the bitstream accordingly.

The smart decoder also makes the same predictions about the subsequent images in the related sequence of images as the encoder. Thus, the encoder can simply send the difference between the prediction and the actual values, thus also reducing the bitstream,

DETAILED DESCRIPTION

1. Introduction/Summary

Compression of digital motion video is the process by which superfluous or redundant information, both spatial and temporal, contained within a sequence of related video frames (frames) is removed. Video compression allows the sequence of frames to be represented by a reduced bitstream, or data flow, while retaining its capacity to be reconstructed in a visually sufficient manner.

Traditional methods of video compression place most of the compression burden, i.e. computational and transmittal, on the encoder, while minimally using the decoder. A tradition video encoder/decoder system requires that the encoder makes all the calculations, inform the decoder of its decisions, then transmit the video data to the encoder along with instructions for reconstruction of each image.

This invention is novel in that it uses a smart decoder to take much of the transmission and instructional burden from the encoder which results in a much smaller

bitstream. Specifically, absent from the bitstream is the information regarding the structural information inherent within the image frame, such as geometry, color, and brightness, which, in a complex frame is a significant amount of video information. Further, absent from the bitstream is information regarding any decision made by the encoder such as segment ordering, segment association and disassociation, etc.

Fig. 1 is an overview drawing of the encoder for use with a compatible decoder as will be described later with respect to Fig. 2. The encoder works as follows:

1. The encoder obtains a reference image frame;
2. The encoder encodes the image frame from step 1;
3. The encoded image from step 2 is reconstructed by the encoder, in the same manner as the decoder will;
4. The encoder segments the reconstructed image from step 4; Alternatively, the encoder segments the original reference image frame from step 1;
5. The segments determined in step 4 are ordered by the encoder, in the same manner as the decoder will;
6. The encoder obtains a new image frame;
7. The motion or kinetic information of each segment, determined in step 4, from the reconstructed, or original image in step 3, to the new image frame in step 6 is determined by motion matching;
8. The encoder encodes the kinetic information;
9. Based on the motion information from step 8, previously hidden regions, also known as the background residue, in the first frame may be exposed in the second frame;

10. The encoder orders the Background residues, in the same manner as the decoder will;
11. The encoder attempts to fill each of the background residues from step 9 and 10.
12. The encoder determines the difference between the predicted fill and the actual fill for each of the background residue areas.
13. The encoder determines the local residue areas in the second image frame, from the segment motion information;
14. The encoder orders the local residues from step 13, in the same manner as the decoder will;
15. The encoder encodes the local residues from step 13.
16. The encoder determines any special instructions associated with the segment information
17. If the image can be reasonably reconstructed primarily from the kinetic information, with assistance from the background residue and the local segment residues, the encoder transmits the following information, and reconstructs the second frame, and continues at step 6:
    a. Flag denoting that the second frame is not a keyframe;
    b. The kinetic information for the segments;
    c. The special instructions for the segments;
    d. The background residue information along with flags denoting coding;
    e. The local residue information along with flags denoting coding;

18. If the image cannot not be reconstructed in relation to the reference frame, the image is encoded as a flag transmitted to inform the decoder, and the encoder continues at step 2.

Fig 2 is an overview drawing of the decoder system with a compatible encoder as described in Fig 1.. The decoder system works as follows:

1. The decoder receives a first encoded image frame from step 3 of the encoder description;
2. The encoded image frame from step 1 is reconstructed by the decoder in the same manner as the encoder;
3. The reconstructed image frame from step 2 is segmented by the decoder. Alternatively, the reconstructed image frame is not segmented by the decoder
4. The decoder receives a flag from the encoder stating whether the second frame from step 19 and 20 of the encoder description is a keyframe, i.e. not represented in relation to any other frame. If so, then the decoder returns to step 1.
5. The decoder receives motion information regarding the segments determined in step 3 from the encoder;
6. The decoder begins to reconstruct a subsequent image frame using the segments obtained in step 3 and motion information obtained in step 4;
7. Based on the motion information from step 4 regarding the segments determined in step 3, the decoder determines where areas, previously hidden, are now revealed, also known as the background residue;
8. The previously background residue locations from step 6 are ordered in the same manner as in the encoder;

9. The decoder attempts to fill the background residue locations from step 6;
10. The decoder receives additional background residue information plus flags denoting the coding method for the additional background residue information from step 8 from the encoder;
11. The decoder decodes the additional background residue information;
12. The computed background residue information and the added background residue information is added to the second image frame.
13. Based on the motion information from step 4 regarding the segments determined in step 3, the decoder determines the location of the local segment residues.
14. The local segment residue locations are ordered in the same manner as the encoder does;
15. The decoder receives coded local segment residue information plus flags denoting the coding method for each local segment residue location;
16. The decoder decodes the local segment residue information;
17. The decoded local segment residue information is added to the second frame.
18. The encoder receives the special instructions, if any, for each segments
19. Reconstruction of the second frame is complete;
20. If there are more frames, the routine continues at step 4

Fig 3 is an overview drawing of the encoder/smart decoder system. The encoder/smart decoder system works as follows:

1. The encoder obtains, encodes and transmits the reference frame;
2. The reference frame from step 2 is reconstructed by both encoder and decoder;

3. Identical segments in the reference frame are determined by both encoder and decoder;
4. The segments from step 3 are ordered in the same way by both the encoder and decoder;
5. The encoder obtains a new image frame;
6. The encoder determines the motion of segments from step 3 by means of motion matching frame from step 5;
7. The encoder encodes motion information;
8. Based on motion information from step 7, the encoder determines previously hidden areas, also known as background residue, which is now exposed in the second frame.
9. The encoder attempts to mathematical predict the image at the background residue regions.
10. The encoder determines if the mathematical prediction was good based upon the difference between the guess and the prediction. The encoder computes additional background residue if necessary.
11. Based on segment information in step 3, and the motion information from step 7, the encoder determines structural information for the local segment residues;
12. Structural information for the local residues from step 11 are ordered by the decoder.
13. Based on the structural information from step 12, regarding the local residues, the encoder encodes the local segment residues.

14. The encoder determines if based upon the kinetic information of the segments, if the second frame should be coded in reference to the first frame. If not, it is coded as a keyframe and the routine begins at step 1.

15. The decoder receives the segment kinetic information from the encoder in step 7.

16. The decoder determines and orders the same background residue at the encoder did in step 8.

17. The decoder makes the identical guess as to the structure of the background residue as the encoder did in step

18. The decoder determines and orders the same local segment residues as determines in step 11 and 12.

19. The decoder receives the local segment residues information from the encoder and flags denoting the coding scheme.

20. The decoder receives the additional background residue information from the encoder.

21. The encoder receives the special information, if any, regarding each segment.

22. Based upon the kinetic information, the local segment residues, and the background residues, both the encoder and decoder identically reconstruct the second frame.

23. The second frame is now the reference frame and the process continues at step 5.

ENCODER WRITE-UP

2. Reference Frame Transmission

Referring to Fig 4, the encoder receives the reference frame, in this case, a picture of an automobile moving left to right with a mountain in the background. The reference frame generally refers to the frame which any other frame is described in relation to.

Fig. 5 is the part of the flow diagram illustrating the procedure by which the encoder initially processes the reference frame. Step 110 begins the process, specifically, the encoder receives the picture described in Fig.4. At step 120, the encoder encodes Fig 4, into a video format, and transmits it to the receptor at step 130. The encoder reconstructs the encoded frame at step 140.

3. Segmentation

Segmentation is the process by which a digital image is subdivided into its component parts, i.e. segments, where each segment represents an area bounded by a radical or sharp change in values within the image.

Persons well versed in the art of computer vision will be aware that segmentation can be done in a plurality of ways. One such way is the watershed method where each pixel is connected to every other pixel in the image frame. As seen in Fig 6, the watershed method segments the image by disconnecting pixels based upon a variety of algorithms. The remaining connected pixels belong to the same segment.

Referring to Fig. 6, At step 210, the encoder segments the reconstructed reference frame to determine the inherent structural features of the image. Alternatively, at step 210, the encoder segments the original image frame for the same purpose. The encoder determines that the segments of Fig. 2 are the car, the wheels, the windows, the street, the sun, and the background. At step 220, the encoder orders the segments based upon a pre-determined criteria and marks them Segments 1 through 8, respectively, as seen in Fig 7.

Segmentation permits the encoder to perform efficient motion matching, motion prediction, and efficient residue coding as explained further in this description.

4. Kinetic Information

Once segmentation has been accomplished, the encoder encodes the kinetic or motion information regarding the movement of each segment.

The kinetic information is determined through a process known as motion matching. Motion matching is the procedure of matching similar regions, often segments, from the first frame to the second frame. At each pixel within a digital image frame, an image is represented by numerical value. Matching occurs when a region in the first frame has identical or near identical pixel values with a region in the second frame.

Generally speaking, a segment is matched with a segment in another frame when the absolute value of the difference in pixel values between the segments is below a pre-determined threshold. While the absolute value of the pixel difference is often used to because it is simple and accounts for negative numbers any number of function would suffice.

In Fig 7a, we see an example of motion matching of a soccer ball between frames 1 and 2. In frame 1, we have a soccer ball, with black and white squares. In frame 2, we have a brownish orange basketball next to the soccer ball. Subtraction of the pixels values contained within the basketball in frame 2 from the soccer ball in frame 1 yield a relatively arbitrary set of non-zero differences. Thus the soccer ball and basketball will not be matched. However, subtraction of the soccer ball in frame 2 from the soccer ball in frame 1 yields a set of mostly zero and close to zero values. Thus the two soccer balls would be considered matched.

The kinetic information transmitted to the decoder can be reduced if related segments can be considered as single groups so that the encoder only needs to transmit one main representative motion vector to the decoder along with motion vector offsets to represent the individual motion of each segment within the group. Grouping is possible if there is previous kinetic information about the segments or if there is multi-scale information about the segments. Multi-scaling will be explained in section 4.2 of the encoder discussion.

Referring to Fig 8, at step 310, the encoder determines if the first frame is a keyframe, i.e. not described in relation to other frames. If the first frame is a keyframe, then there isn't any previous kinetic information and grouping is only possible if there is multi-scale information regarding the image frame. However, if the first frame is not a keyframe, then there will be some previous kinetic information to group segments. Therefore, if the first frame is not a keyframe, step 320, will execute the motion grouping routine, described here as section 4.1.

However, if the first frame is a keyframe, then step 310, goes to step 330, where the encoder determines if there is any multi-scale information available to it. If there is, then step 340 executes the Multi-scaling routine in section 4.2, otherwise at step 350, the encoder decides not to group any segments.

If the first frame is a keyframe, and thus previous kinetic information is not available, and there is no multi-scale information available either, the encoder cannot group the segments and then, at step 350, encoder determines that it cannot group any segments together.

4.1 Motion Vector Grouping

Motion vector grouping only occurs when there is previous motion information so that the encoder can determine which segments to associate. Motion vector grouping begins at step 510 in Fig 10, where the previous motion vector of each segment is considered. Segments which exhibit similar motion vectors are grouped together at step 520. At step 530, the motion vector for the group is determined by combining the motion vectors within the groups. Thus, for each segment within the group, only the motion vector difference, i.e. the difference between the segment's motion vector and the characteristic motion vector will be eventually transmitted. (See step 540) One example of a characteristic motion vector would be an average motion vector.

At step 550, the encoder orders the groups. However, before the motion information can be transmitted, further reduction might occur through motion prediction at step 555, described here in section 4.1.1. Once the motion information is determined it is stored at step 560.

4.1.1 Motion Prediction

Referring to Fig 11, at step 610, the encoder considers a segment. At step 620, the encoder determines if there is previous motion information for the segment so that its motion can be predicted. If there isn't any previous motion information, the encoder chooses the next segment and continues.

If there is previous motion information the encoder predicts the motion of the segment at step 630 and compares its prediction to the actual motion of the segment at step 640. The motion vector offset is initially predicted at step 650 as a function of the actual and predicted motion vectors. An example of a motion vector calculation would be the difference between the actual and predicted motion vectors. At step 660 the

encoder makes the final calculation for the motion vector offset. An example of the final motion vector calculation could be the difference between the initial motion vector and the characteristic motion vector.

At step 670, the encoder determines if there are any more segments, if so, then at step 680, the encoder considers the next segment and continues at step 620. Otherwise the prediction routine ends.

4.2 Multi-Scale Grouping

Multi-scaling grouping is an alternative to grouping segments by previous motion. Moreover, multi-scaling in may used in conjunction with motion grouping. Multi-scaling is the process of creating lower resolution versions of an image. An example of creating multiple scales is through the repeated application of a smoothing function. The result of creating lower resolution images is that as the resolution decreases, only larger, more dominant features remain visible. Thus for example, the stitching on a football may become invisible at lower resolutions, yet the football itself remains discernible

An example of the multi-scale processes is as follows: referring to Fig. 9 at step 410, the encoder considers the coarsest image scale (i.e. lowest resolution) for the frame and at step 420 determines which segments have remained visible. The coarsest image scale is used because at that point, only the absolute largest, most dominant features remain, usually corresponding to the outline of major objects remain visible. While smaller, less dominant segments are no longer discernible at the lower resolutions. At step 430, invisible segments which are wholly contained within a given visible segment are associated with the segment and considered one group. This is because the smaller, now invisible segments are often share a relationship with the larger object and will likely

have similar kinetic information. A decision is made at step 440. If there are more visible segments, at step 450, the encoder considers the next segment and continues at step 430. Otherwise the Multi-scaling grouping process ceases.

5. Residue Coding

Referring to Figs. 12-15, the residue is the portion of the image left over after the structural information has been moved. Residue falls under two classifications; new information and local residues.

5.1 New information

As shown in Fig. 12, as the segment moves, previously hidden or obstructed areas may become visible for the first time. In Fig. 12, three regions become visible as the car moves. They are the area behind the back of the car and the two areas behind the wheels. These are marked regions 1 through 3, respectively. Referring to Fig 13, at step 710, the encoder determines where the previously obstructed image regions occur. At step 720, the encoder orders the region using a predetermined ordering system. Using the information surrounding the regions, the encoder makes a mathematical guess as to the structure of the regions. Yet, the encoder also knows precisely what images were revealed at these regions. Thus at Step 740, the encoder considers a region and determines if the mathematical prediction was sufficient by comparing the guess with the actual image. If the prediction was not close, at step 770, the encoder will encode the region or the difference and store the encoded information with a flag denoting the coding mechanism. Otherwise, if the guess was close enough, the encoder stores a flag denoting that fact at step 745.

At step 750, the encoder determines if there are any more newly unobstructed regions. If so the next region is considered and the routine continues at step 730, else it ceases at step 799.

5.2 Local residues

Referring to Fig. 14, the local residue is the portion of the image in the neighborhood of a segment, left over after the segments have been moved, i.e. the car and mountain appear smaller in the subsequent frame. The structure of the residue will depend on how different the new segments are from the previous segments. It may be a well-defined region, or set of regions, or it may be patchy. Different types of coding methods are ideal for different types of local residue. Since the decoder knows the segment motion, it knows where most of the local residues will be located.

Referring to Fig 15, at step 810, the encoder determines the locations of the local residues and orders the regions where the local residues occurs using a pre-determined ordering scheme at section 820. At step 830, the encoder considers the first local residue, and makes a decision as the most efficient method of coding it and encodes it at step 840. The encoder stores a flag denoting the coding mechanism as well as the coded residue at step 850. If there are more local residue locations, step 860 will consider the next local residue location and continue at step 840, otherwise the at step 870, the encoder executes the keyframe routine at Fig 15a, step 880.

Referring to Fig 15a, at step 880, the encoder determines if the second frame should be coded as a keyframe. If yes, then step 885, the encoder discards the kinetic information, the background residue, and the local segment residues and continues at step 120. Otherwise, the routine transmits the kinetic information, the background residue, and the local segment residues to the decoder at step 890.

6. Special Commands

The encoder transmits embedded commands and instructions regarding each segment into the bitstream as necessary. Examples of these commands include, but are not limited to, getting static web pages, obtaining another video bitstream, waiting for text, etc.

The encoder can embed these commands at any point within the bitstream subsequent to the decoder ordering the segments. Fig 14a, is an example of one point where the commands are be embedded within the data stream.

Referring to Fig 14a, at step 1610, the encoder considers the first segment. At step 1620, it transmits a special instruction. At step 1630, the encoder determines if there are any special instructions for the segment. If yes, then at step 1640, the instructions are transmitted to the decoder and at step 1650 the encoder determines if there are any more segments. If there are no special instructions associated with the segment, the encoder proceeds directly to step 1650. If there are more segments, at step 1660, the encoder considers the next segments are continues to step 1620, otherwise the routine ends at step 1699.

DECODER DESCRIPTION

2. Reference Frame Reception

Referring to Fig 16, the decoder receives the encoded reference frame of a picture of an automobile moving left to right with a mountain in the background (See Fig. 4). The reference frame generally refers to the frame which other, subsequent frames are described in relation to.

Fig. 16 illustrates the flow diagram of the above process. Step 910 begins the process where the decoder receives an encoded image frame. At step 920, the decoder reconstructs the encoded image frame.

At step 930, the decoder receives a keyframe flag. This flag denotes whether the second frame is a keyframe or can it be reconstructed from the kinetic and residue information. If the second frame is a keyframe, then the decoder returns to step 910, where it received the keyframe as a first frame, otherwise the routine continues.

3. Segmentation

As previously described, segmentation is the process by which a digital image is subdivided into its components parts, i.e. segments, where each segment represents an area bounded by a radical or sharp change in values within the image.

Referring to Fig 17, at step 1010, the decoder segments the reconstructed reference frame to determine the inherent structural features of the image. The decoder determines that the segments in Fig. 4 are the car, the wheels, the doors, the windows, the street, the mountain and the background. At step 1020, the decoder will order the segments based upon the same predetermined criteria as the encoder and mark the segments as 1 through 10 as seen in Fig 7.

4. Kinetic Information

Once segmentation has been accomplished, the decoder receives a keyframe flag from the encoder. This flag tells the encoder if the first frame is a keyframe. The decoder receives the kinetic information regarding the movement of each segment. The kinetic information tells the decoder the position of the segment in the new frame relative to its position in the previous frame. The kinetic information is reduced if the segments with related motion can be grouped together and represented by one motion vector. The

kinetic information received by the decoder depends on several factors: to wit; 1) the reference frame is a key frame, and 2) if not, is multi-scaling information available.

Referring to Fig. 18, at step 1110, the decoder determines if the reference frame is a keyframe, i.e. a frame not defined in relation to any other frame. If so, then there is no previous motion information for potential grouping of segments, therefore the decoder attempts to use multi-scale information for segment grouping, if available. At step 1120, the decoder determines if there is multi-scale information available.. If the first frame is a keyframe and there is multi-scale information available to the decoder, the decoder will initially group related segments together using the multi-scale routine executed at step 1130, and described in section 4.1 of the description. Conversely, if there is no multi-scale information available for the first frame, then at step 1150, the motion vectors are transmitted by the encoder and received by the decoder.

However, at step 1110, if the decoder determines that the first frame is not the keyframe, then it executes the motion grouping routine at step 1140, and described in section 4.2. Alternatively, it may use the multi-scale grouping described in step 4.1

4.1 Multi-Scale Grouping

Multi-scale grouping only occurs when the first frame is a keyframe and there is multi-scale information available to the decoder.

Referring to Fig. 19 at step 1210, the decoder considers the coarsest image scale for the frame and at step 1220 determine which segments have remained visible. At step 1230, invisible segments which are wholly contained within the a given visible segment are associated with the segment. A decision is made at step 1240. If there are more visible segments, at step 1260, the decoder considers the next segment and continues at

step 1230. Otherwise the Multi-scaling grouping process receives the motion vectors and motion vector offsets for the segments then ceases.

4.2 Motion Vector Grouping

Referring to Fig 20, at step 1310, the decoder considers a segment. At step 1320, the encoder determines if there is previous motion information for the segment so that its motion can be predicted. If there isn't any previous motion information, the encoder chooses the next segment and continues.

If there is previous motion information the encoder predicts the motion of the segment at step 1330 and receives the motion vector prediction correction at step 1340.

At step 1350, the encoder determines if there are any more segments, if so, then at step 1360, the encoder considers the next segment and continues at step 1320. Otherwise the prediction routine ends.

5. Residue Coding

The residue is the portion of the image left over after the structural information has been moved. Residue falls under two classifications; background and local residues.

5.1 Background residue

As shown in Fig 12, as the car moves, previously hidden or obstructed areas may become visible for the first time. The decoder knows where these areas are and orders them using a predetermined ordering scheme. In Fig 12. Three regions become unobstructed, specifically, behind the car, and behind the two wheels. These regions are marked Regions 1 through 3, as seen in Fig 12.

Referring to Fig 21, at step 1410, the decoder considers the background residue regions and orders the regions at step 1420. At step 1430, it makes a mathematical prediction on the structure of the first background residue location. At step 1440, the

decoder receives a flag denoting how good the prediction was and if correction is needed. Step 1450 makes a decision, if the prediction is sufficient, the routine continues at step 1470, otherwise at step 1460, receives the encoded region and the flag denoting the coding scheme and reconstructs as necessary. If there are more background residue locations, at step 1470, the decoder, at step 1480, considers the next region and continues at step 1430. Otherwise the decoder goes to step 1490 where reconstruction continues and the process ceases.

5.2 Local residues

Referring to Fig 15, as previously explained, the local segment residue is the portion of the image, in the neighborhood of the segment, left over after the segment has been moved, i.e. the car and the mountain appear smaller in the subsequent frame. Also, as explain before, the structure of the local residue may be varied. The decoder knows that most of the local residues will appear around the segments.

Referring to Fig. 23, at step 1510, the decoder considers the first segment. At step 1520, the decoder receives a flag denoting the coding method and receives the encoded local residue for that segment. Step 1530 determines if there are any more segments and if not end at 1590 where reconstruction concludes. Otherwise at step 1540 the decoder considers the next segment and continues at step 1520. The routine ends at step 1599.

6. Special instructions

In addition to structural information regarding the image frame, the decoder is capable of receiving and executing commands embedded within the bitstream and associated with the various segments. As before, because the encoder and decoder are synchronized and are working with the same reference frame, the encoder is not required to transmit the structural information associated with the commands. The embedded

commands are held in abeyance until a user-driven event, i.e. a mouseclick, occurs. Fig 24 is an example of one potential way to embed the commands.

Referring to Fig 24, at step 1710, the decoder considers the first segment, at step 1720 it received a special instruction flag. The decoder determines, at step 1730, if there are special instructions or commands associated with the segment. If so, the decoder receives the commands at step 1740. At step 1750, the decoder determine if there are any more segments. If there were no special instructions or commands, the decoder goes to step 1750 directly.

If there are more segments, the decoder, at step 1760, considers the next segment and continues at step 1720, otherwise the routine ends at step 1799.

Referring to Fig 25, at step 1810, the decoder determines if the user-driven event has occurred. If it has, the decoder determines which segment the user-driven event refers to at step 1820. At step 1830, the associated command is executed. The decoder proceeds to step 1840. If the user-driven event has not occurred, the routine proceeds directly to step 1840. At step 1840, if the termination command has been sent, the routine exits at step 1899, otherwise the routine continues at step 1810.

6. Reconstruction

The second frame is reconstructed into a video format based upon the kinetic motion of the segments, and local segment residues and the background residues.

Video format

The description in the previous sections titled encoder and decoder description defines a specific new video format.

WHAT IS CLAIMED IS:

1. A method of transmitting video information comprising:

(a) obtaining a first video frame containing image data;

(b) obtaining structural information inherent in said image data;

(c) obtaining a second video frame to be encoded relative to said first video frame;

(d) computing kinetic information for describing said second video frame in terms of said structural information of said first video frame; and

(e) transmitting said kinetic information to a decoder for use in reconstructing said second video frame based on said decoder's generation of said structural information of said first video frame.

ENCODER DESCRIPTION

1	The encoder obtains a reference image frame.
2	The encoder encodes the image frame from step 1 and transmits it to the decoder.
3	The encoded image from step 2 is reconstructed by the encoder, in the same manner as the decoder will;
4	The encoder segments the reconstructed image from step 4 ; Alternatively, the encoder segments the original reference image frame from step 1;
5	The segments determined in step 4 are ordered by the encoder, in the same manner as the decoder will;
6	The encoder obtains a new image frame;
7	The motion or kinetic information of each segment, determined in step 4, from the reconstructed, or original image in step 3, to the new image frame in step 6 is determined by motion matching;
8	The encoder encodes the kinetic information;
9	Based on the motion information from step 8, previously hidden regions, also known as as the background residue, in the first frame may be exposed in the second frame;
10	The encoder orders the Background residues, in the same manner as the decoder will;
11	The encoder attempts to fill each of the background residues from step 9 and 10:
12	The encoder determines the difference between the predicted fill and the actual fill for each of the background residue areas.
13	The encoder determines the local residue areas in the second image frame, from the segment motion information;
14	The encoder orders the local residues from step 13, in the same manner as the decoder will;
15	The encoder encodes the local residues from step 13.
16	If the image can be reasonably reconstructed primarily from the kinetic information, with assistance from the background residue and the local segment residues, the encoder transmits the following information, and reconstructs the second frame, and continues at step 6: a. Flag denoting that the second frame is not a keyframe b. The kinetic information for the segments c. The background residue information along with flags denoting coding The local residue information along with flags denoting coding
17	If the image cannot be reconstructed in relation to the reference frame, the image is encoded as a flag transmitted to inform the decoder, and the encoder continues at step 2.

FIG.1

DECODER DESCRIPTION

1	The decoder receives a first encoded image frame from step 3 of the encoder description;
2	The encoded image frame from step 1 is reconstructed by the decoder in the same manner as the encoder;
3	The reconstructed image frame from step 2 is segmented by the decoder. Alternatively, the reconstructed image frame is not segmented by the decoder
4	The decoder receives a flag from the encoder stating whether the second frame from step 19 and 20 of the encoder description is a keyframe, i.e. not represented in relation to any other frame. If so, then the decoder returns to step 1.
5	The decoder receives motion information regarding the segments determined in step 3 from the encoder;
6	The decoder begins to reconstruct a subsequent image frame using the segments obtained in step 3 and motion information obtained in step 4;
7	Based on the motion information from step 4 regarding the segments determined in step 3, the decoder determines where areas, previously hidden, are now revealed, also known as the background residue;
8	The previously background residue locations from step 6 are ordered in the same manner as in the encoder;
9	The decoder attempts to fill the background residue locations from step 6;
10	The decoder receives additional background residue information plus flags denoting the coding method for the additional background residue information from step 8 from the encoder;
11	The decoder decodes the additional background residue information;
12	The computed background residue information and the added background residue information is added to the second image frame.
13	Based on the motion information from step 4 regarding the segments determined in step 3, the decoder determines the location of the local segment residues.
14	The local segment residue locations are ordered in the same manner as the encoder does;
15	The decoder receives coded local segment residue information plus flags denoting the coding method for each local segment residue location;
16	The decoder decodes the local segment residue information;
17	The decoded local segment residue information is added to the second frame.
18	Reconstruction of the second frame is complete;
19	If there are more frames, the routine continues at step 4.

FIG 2.

ENCODER/DECODER SYSTEM

	Encoder		Decoder
1	Obtain encode, transmit frame	→	Receive Frame
2	Reconstruct Frame		Reconstruct Frame
3	Segmentation		Segmentation
4	Order segments		Order Segments
5	Obtain new image frame		
6	Determine segment motion		
7	Encode motion information		
8	Determine background residue		
9	Predict background residue fill		
10	Determine sufficiency of prediction		
11	Determine local residue		
12	Order local residue locations		
13	Encode residue		
14	Is 2nd frame keyframe, yes, goto 5	→	Receive Keyframe Flag
15	Transmit motion data	→	Receive motion data
16			Determine and order background residue
17			Predict background residue
18	Transmit background residue data	→	Receive additional background residue data
19			Determine and order local segment residues
20	Transmit local segment residue	→	Receive local segment residue
21	Reconstruct 2nd frame		Reconstruct 2nd frame
22	Goto Step 5		Goto Step 5

FIG 3.

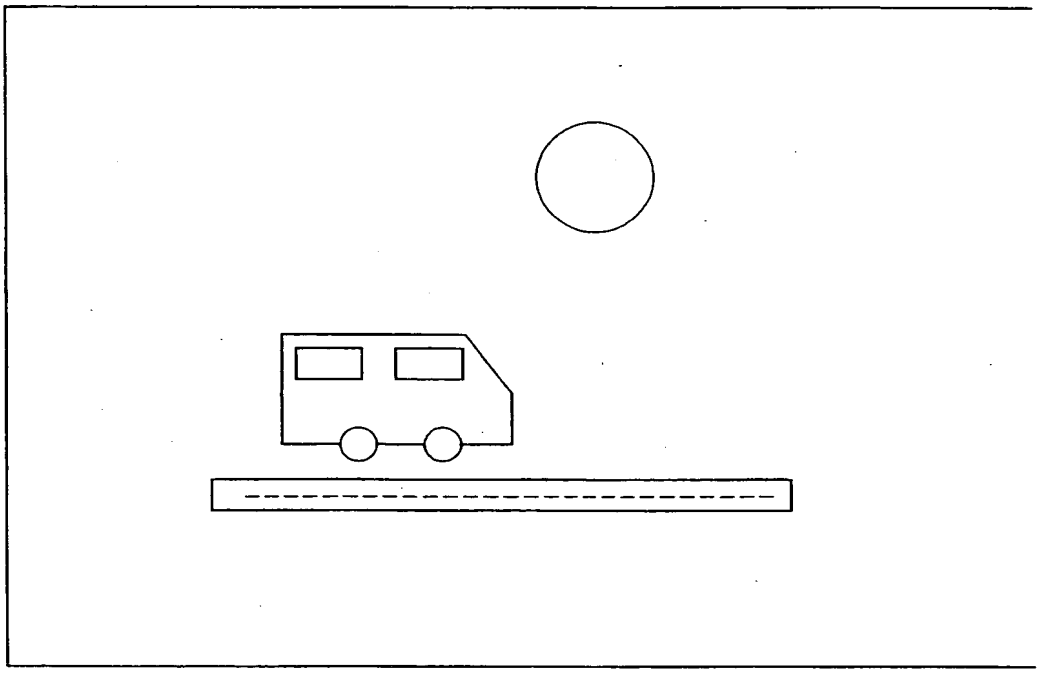

FIG. 4

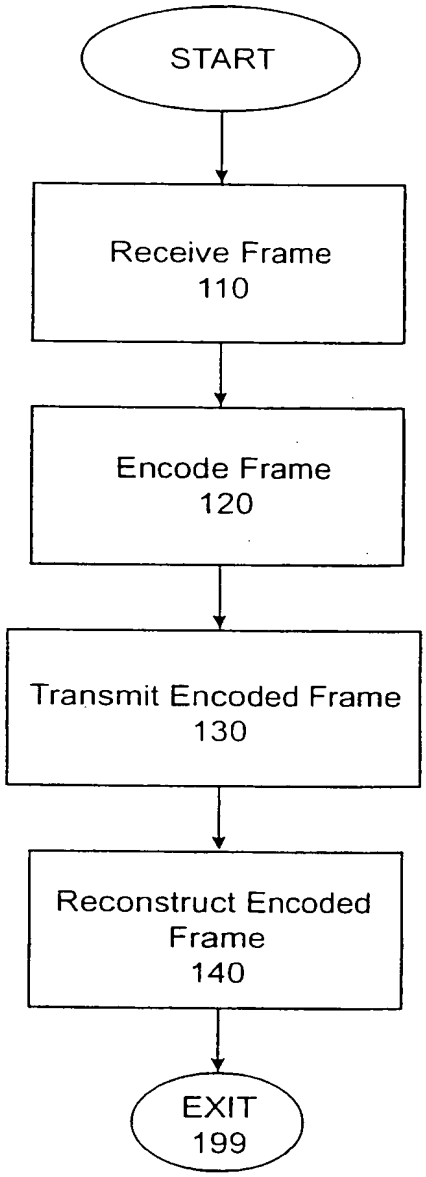
START
Receive Frame
110
Encode Frame
120
Transmit Encoded Frame
130
Reconstruct Encoded
Frame
140
EXIT
199

FIG. 5

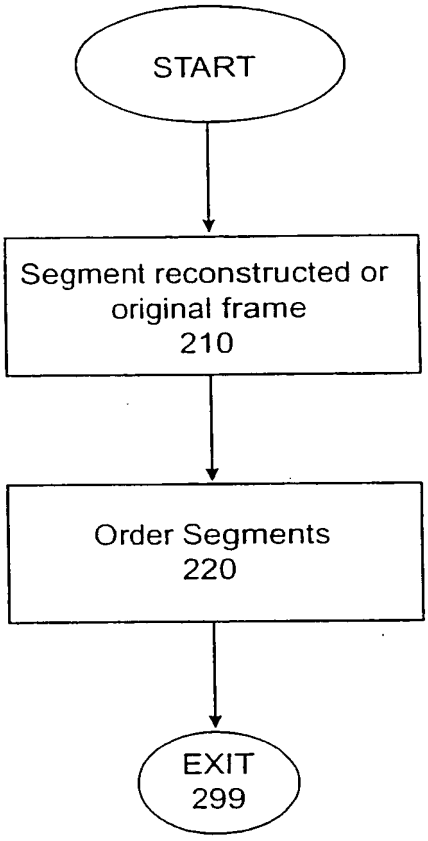
START
Segment reconstructed or original frame 210
Order Segments 220
EXIT 299

FIG. 6

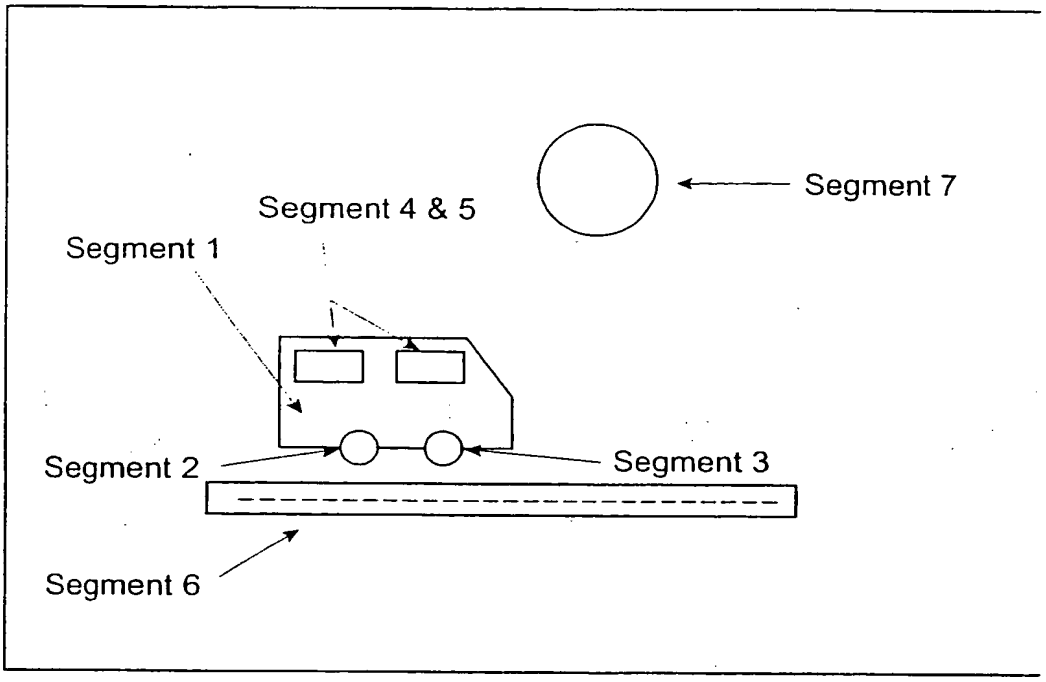

FIG. 7A

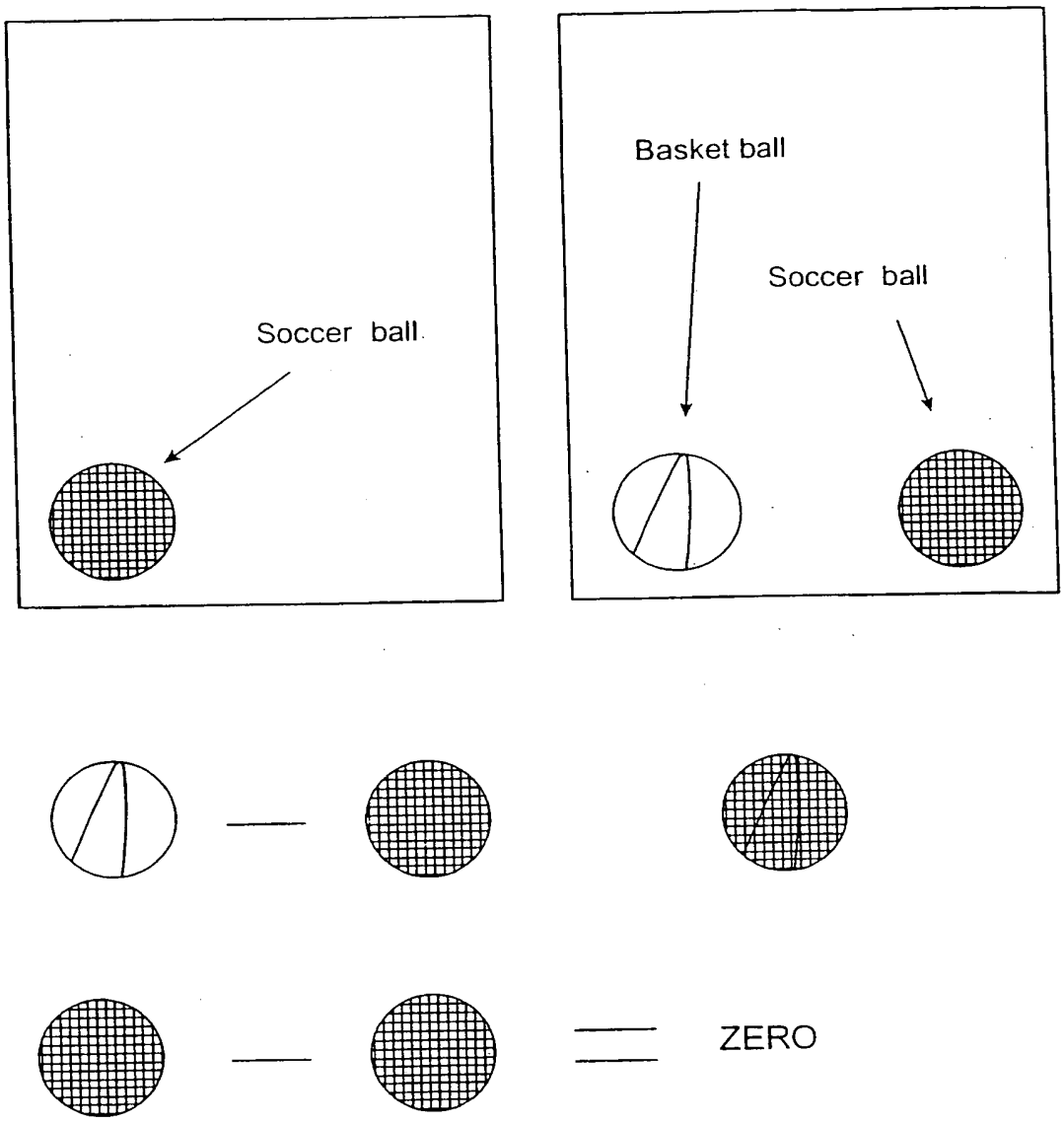

FIG. 7B

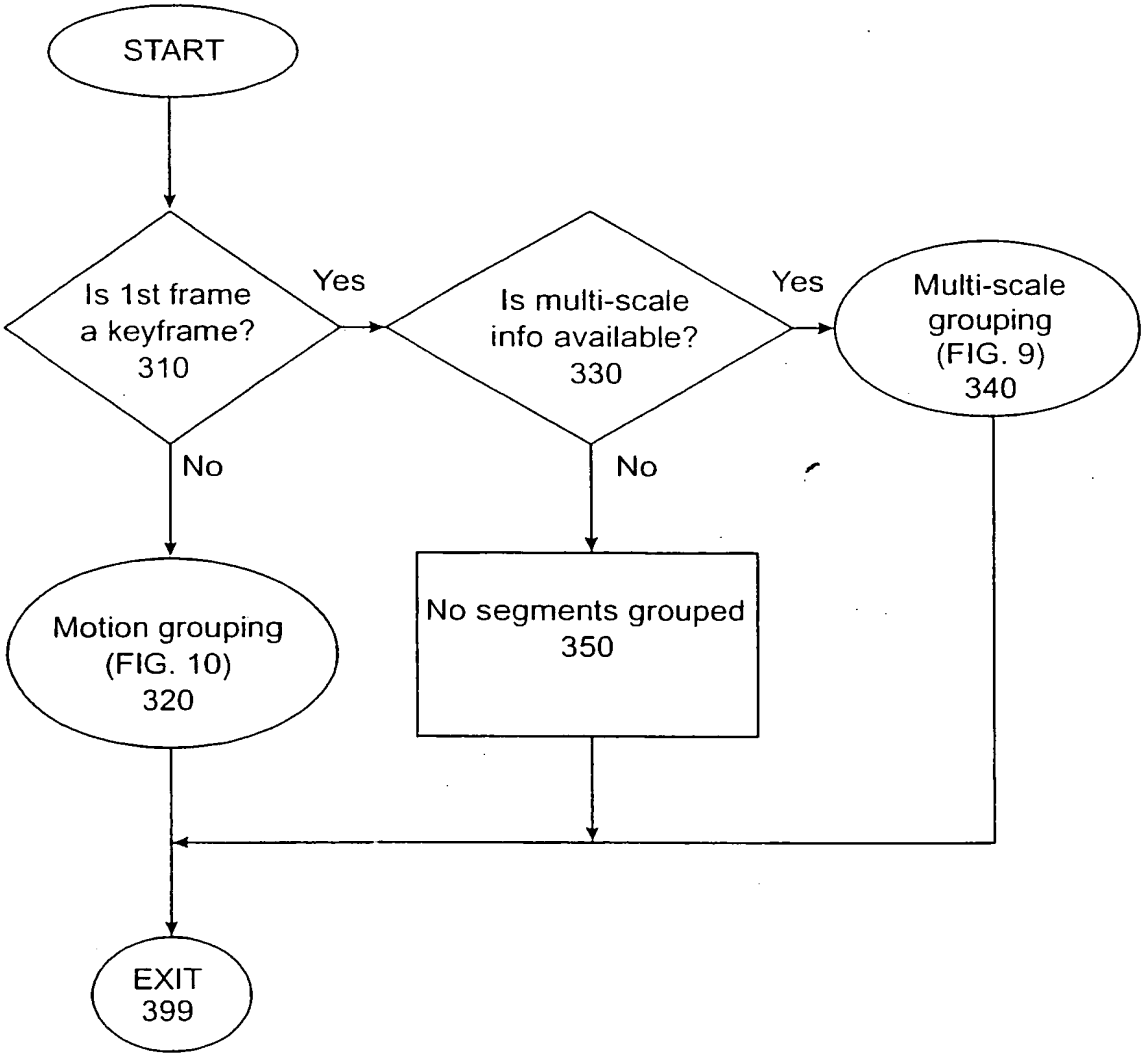
START
Is 1st frame a keyframe? 310
Yes
Is multi-scale info available? 330
Yes
Multi-scale grouping (FIG. 9) 340
No
No
Motion grouping (FIG. 10) 320
No segments grouped 350
EXIT 399

FIG. 8

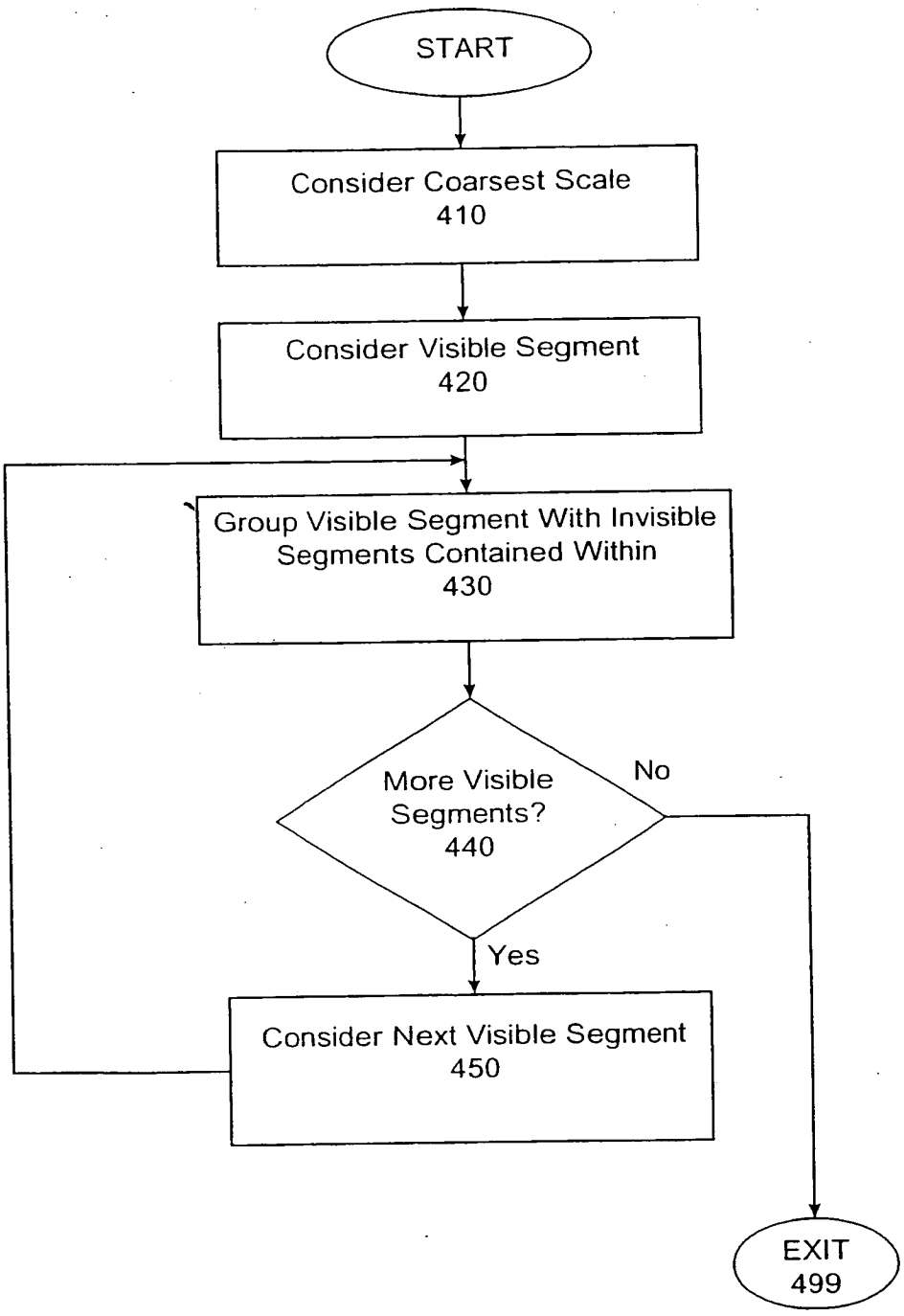
START
Consider Coarsest Scale
410
Consider Visible Segment
420
Group Visible Segment With Invisible Segments Contained Within
430
More Visible Segments?
440
No
Yes
Consider Next Visible Segment
450
EXIT
499

FIG. 9

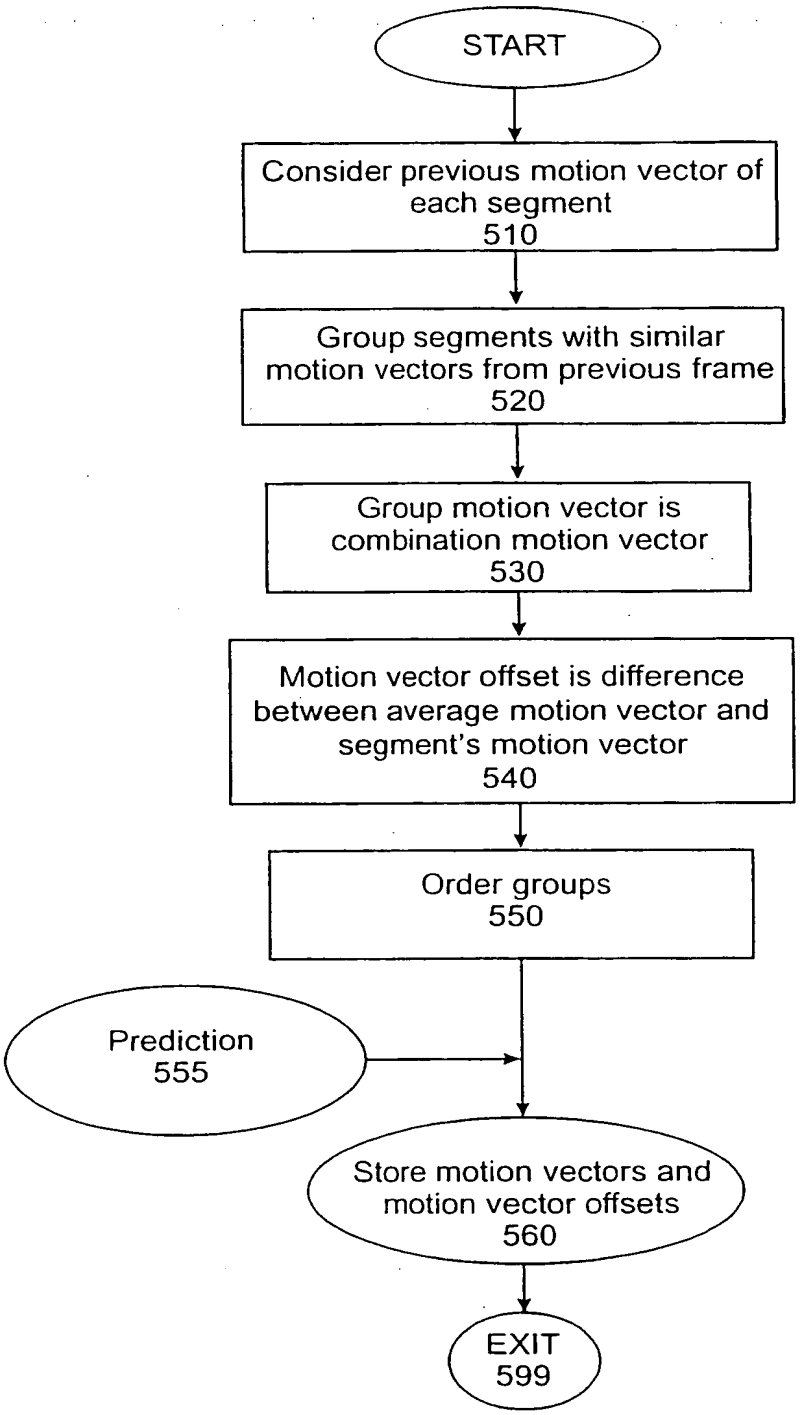
START
Consider previous motion vector of each segment
510
Group segments with similar motion vectors from previous frame
520
Group motion vector is combination motion vector
530
Motion vector offset is difference between average motion vector and segment's motion vector
540
Order groups
550
Prediction
555
Store motion vectors and motion vector offsets
560
EXIT
599

FIG. 10

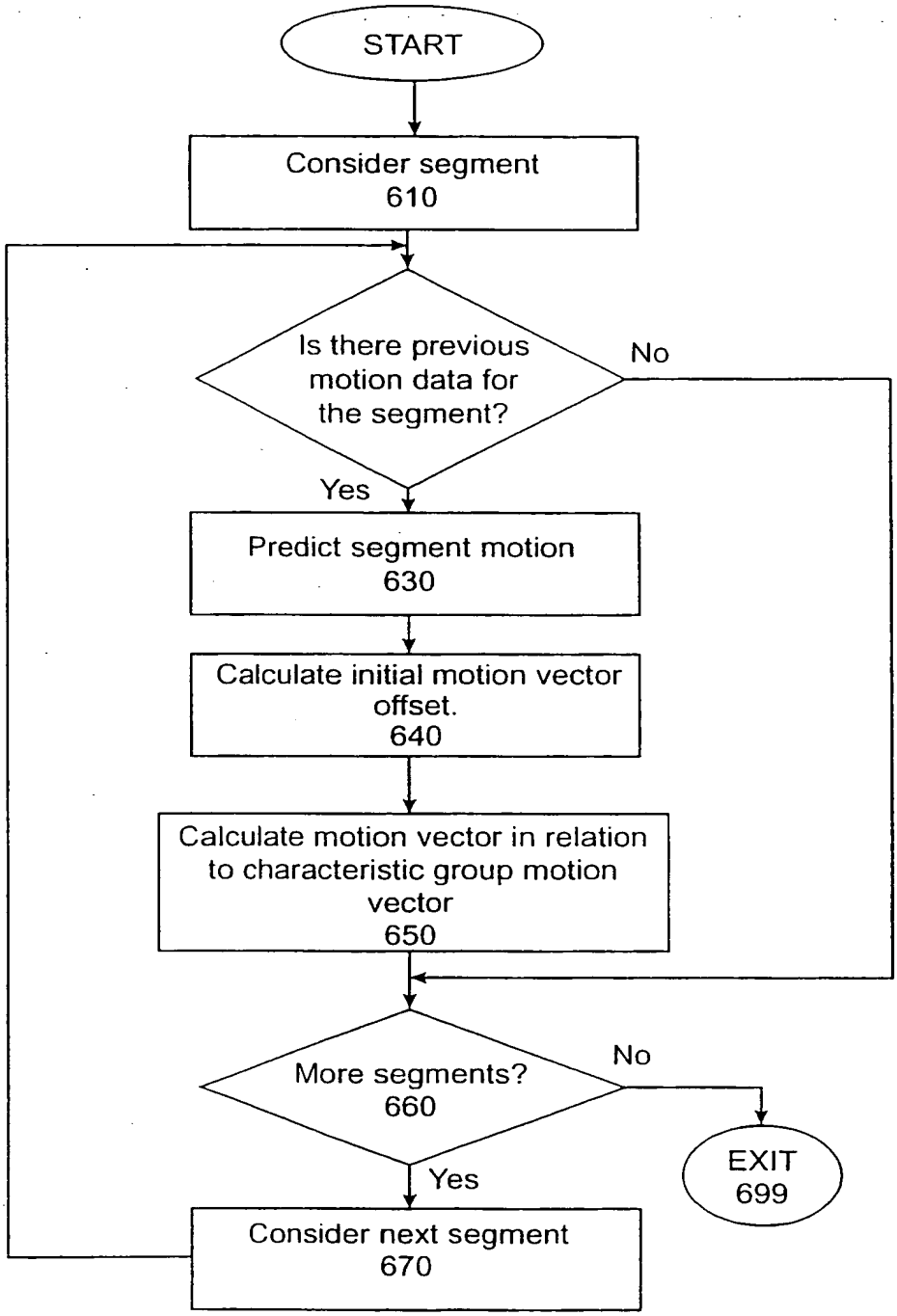
START
Consider segment
610
Is there previous motion data for the segment?
No
Yes
Predict segment motion
630
Calculate initial motion vector offset.
640
Calculate motion vector in relation to characteristic group motion vector
650
More segments?
660
No
Yes
EXIT
699
Consider next segment
670

FIG. 11

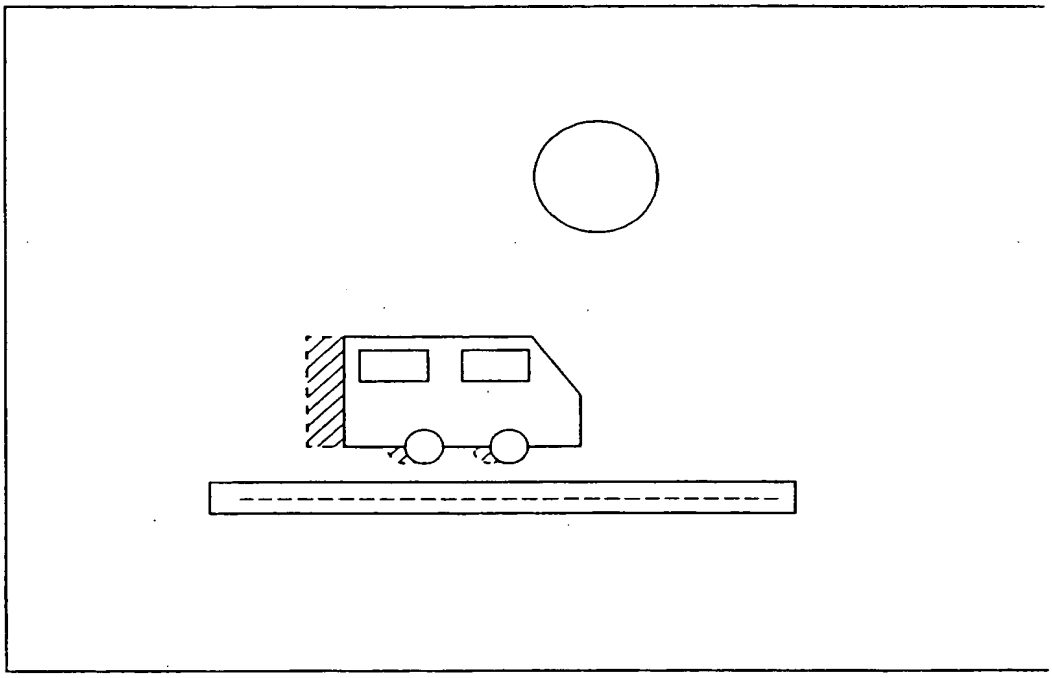

FIG. 12

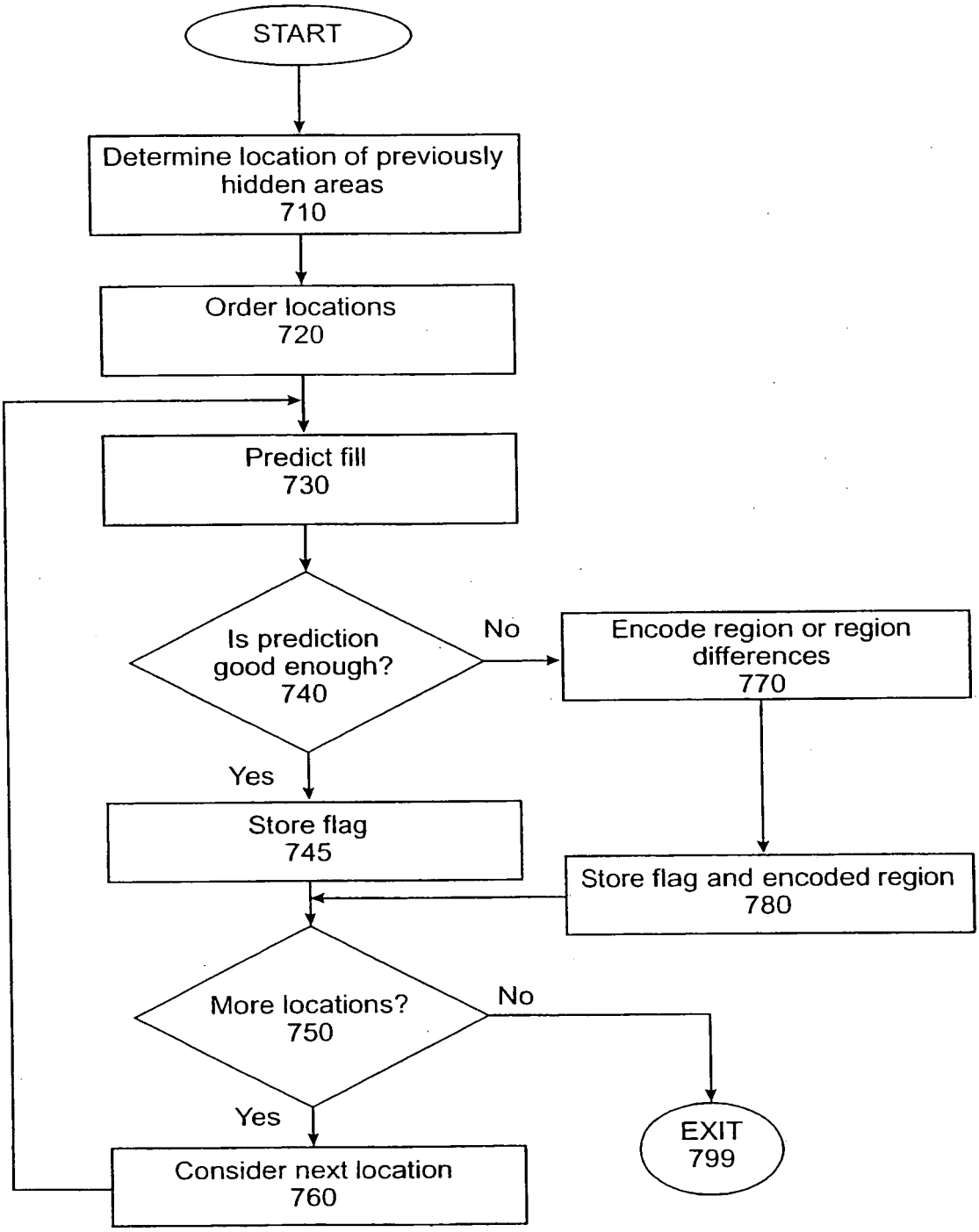
START
Determine location of previously hidden areas 710
Order locations 720
Predict fill 730
Is prediction good enough? 740
No
Encode region or region differences 770
Yes
Store flag 745
Store flag and encoded region 780
More locations? 750
No
Yes
EXIT 799
Consider next location 760

FIG. 13

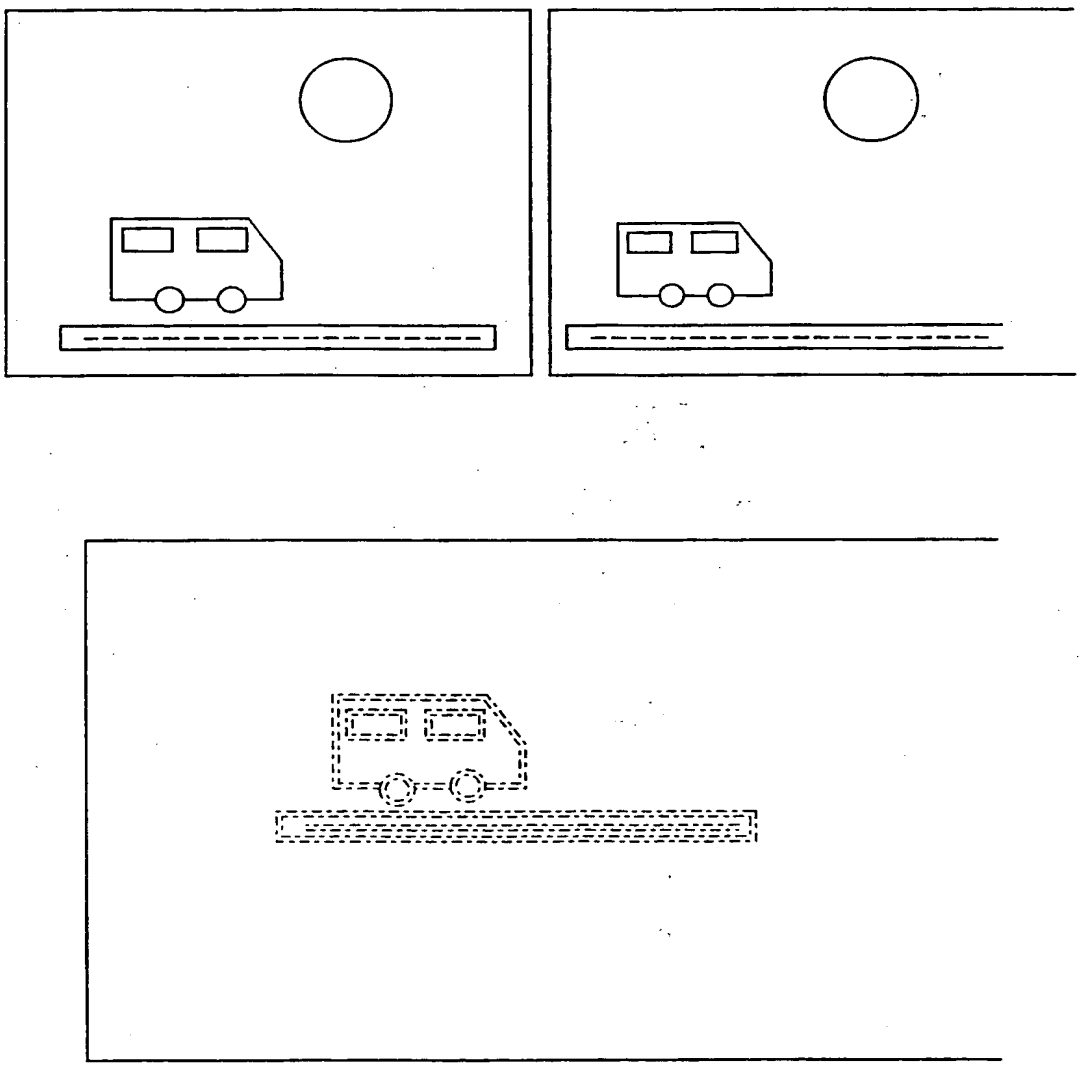

FIG. 14

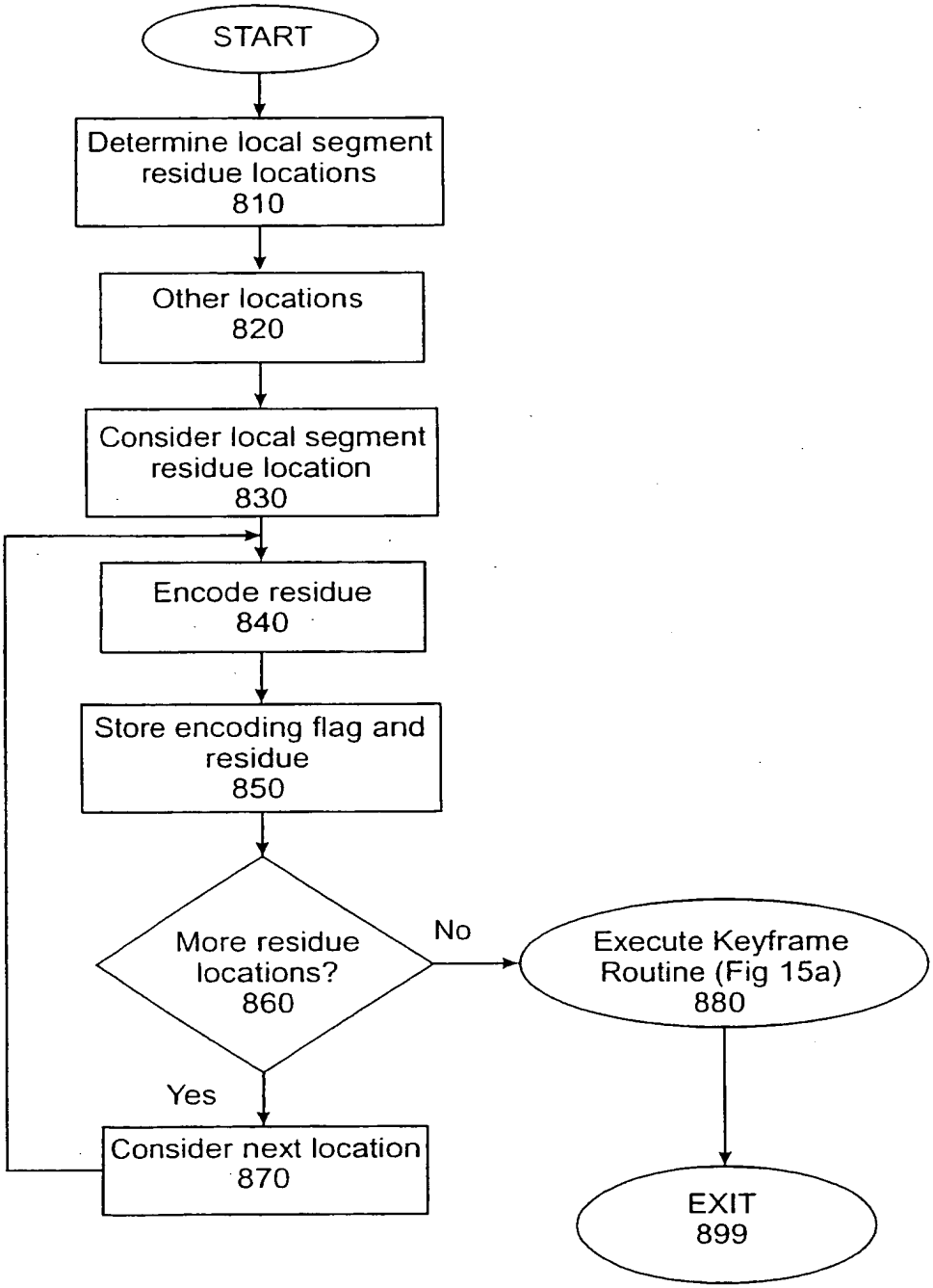
START
Determine local segment residue locations 810
Other locations 820
Consider local segment residue location 830
Encode residue 840
Store encoding flag and residue 850
More residue locations? 860
No
Execute Keyframe Routine (Fig 15a) 880
Yes
Consider next location 870
EXIT 899

FIG. 15A

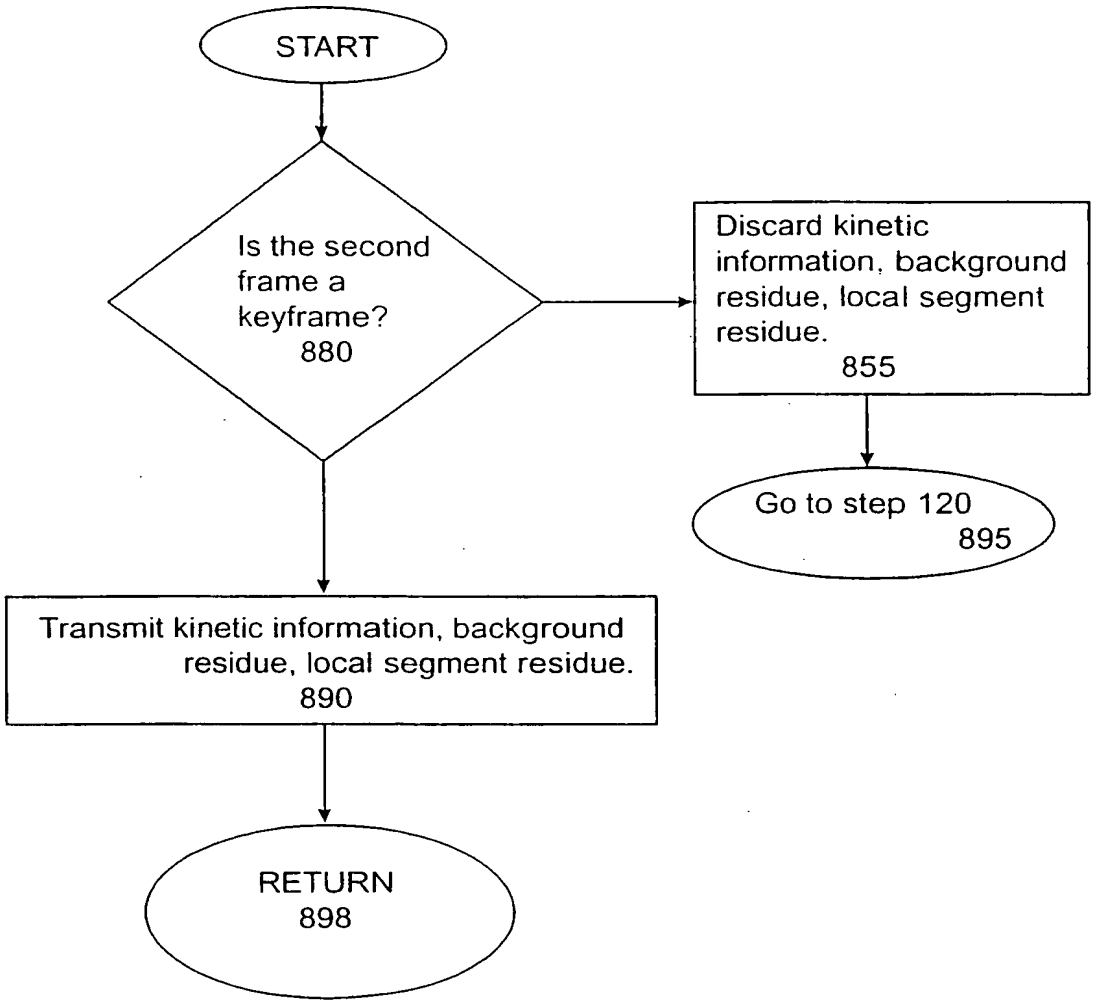
START
Is the second frame a keyframe?
880
Discard kinetic information, background residue, local segment residue.
855
Go to step 120
895
Transmit kinetic information, background residue, local segment residue.
890
RETURN
898

FIG. 15B

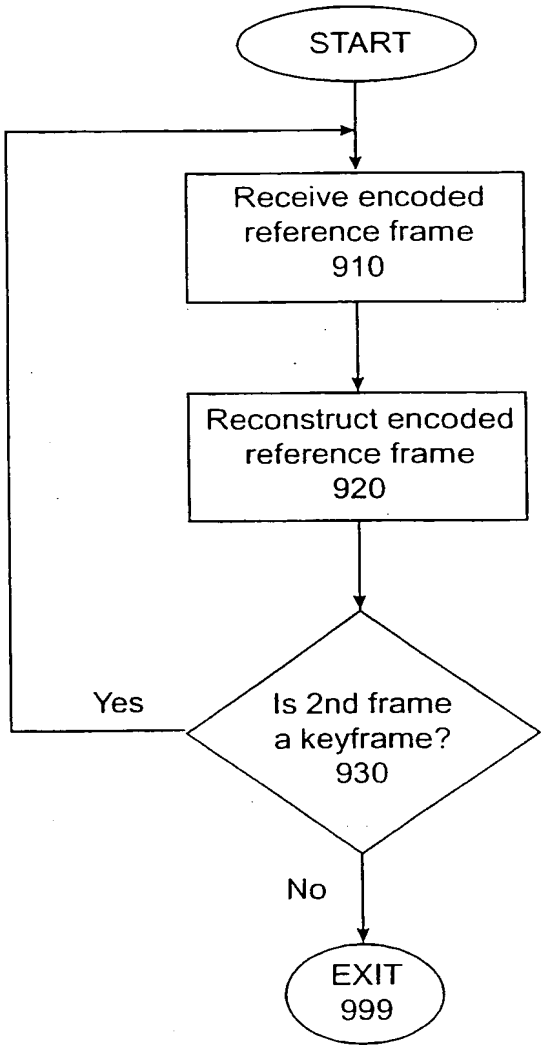
START
Receive encoded reference frame 910
Reconstruct encoded reference frame 920
Is 2nd frame a keyframe? 930
Yes
No
EXIT 999

FIG. 16A

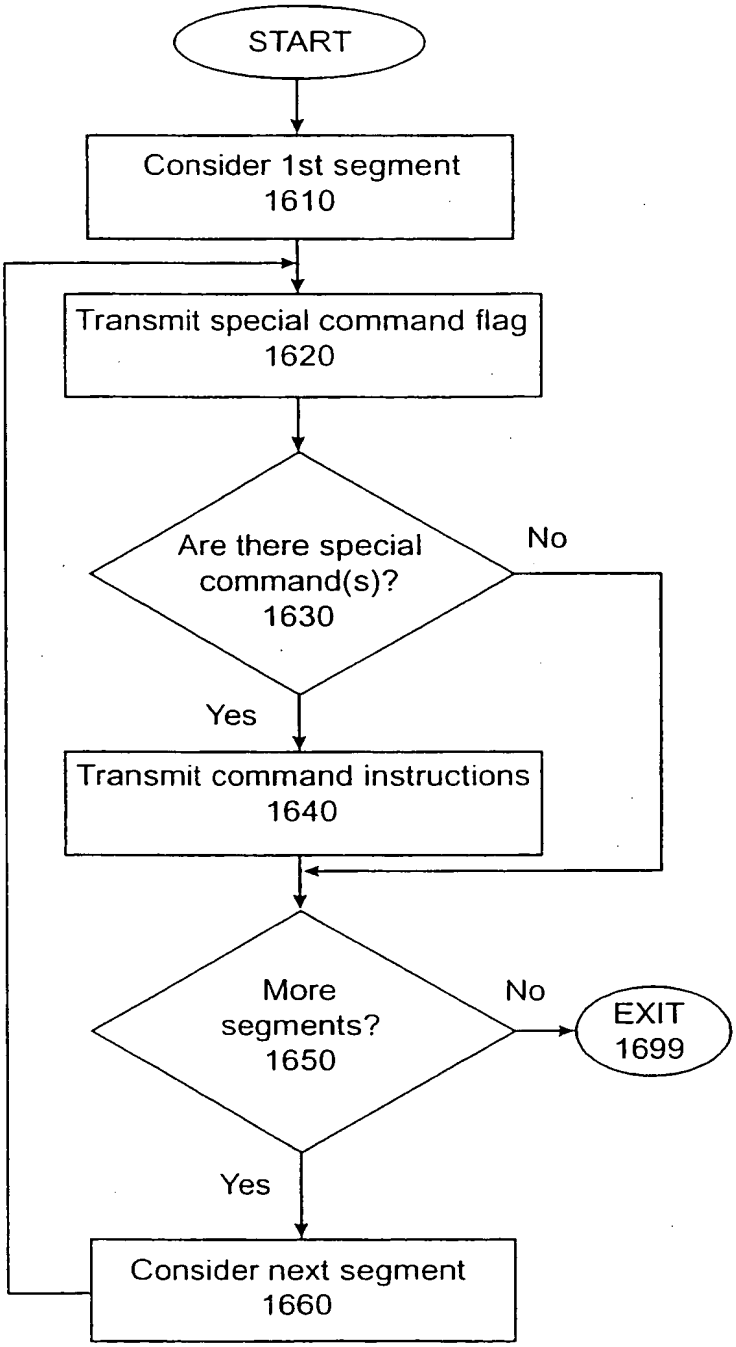
START
Consider 1st segment
1610
Transmit special command flag
1620
Are there special command(s)?
1630
No
Yes
Transmit command instructions
1640
More segments?
1650
No
EXIT
1699
Yes
Consider next segment
1660

FIG.16B

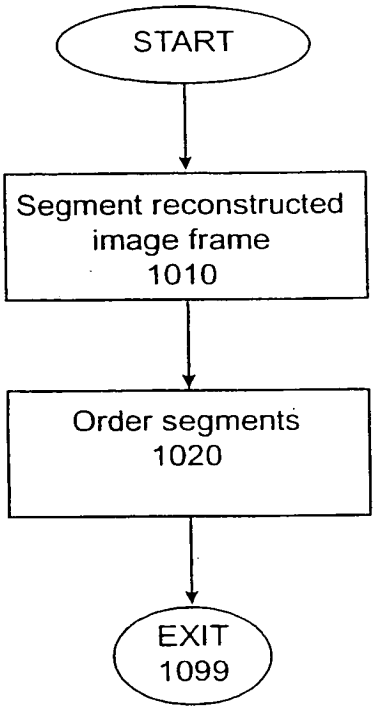
START
Segment reconstructed image frame 1010
Order segments 1020
EXIT 1099

FIG. 17

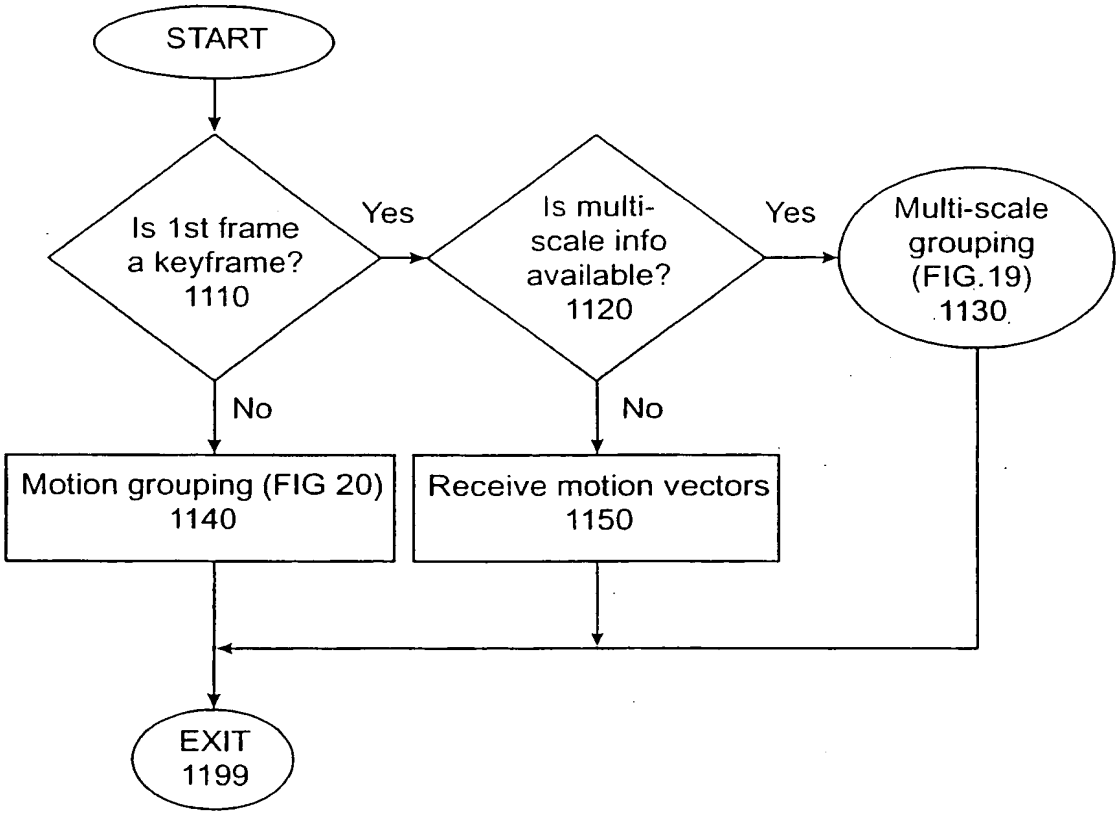
START
Is 1st frame a keyframe? 1110
Yes
Is multi-scale info available? 1120
Yes
Multi-scale grouping (FIG.19) 1130
No
No
Motion grouping (FIG 20) 1140
Receive motion vectors 1150
EXIT 1199

FIG. 18

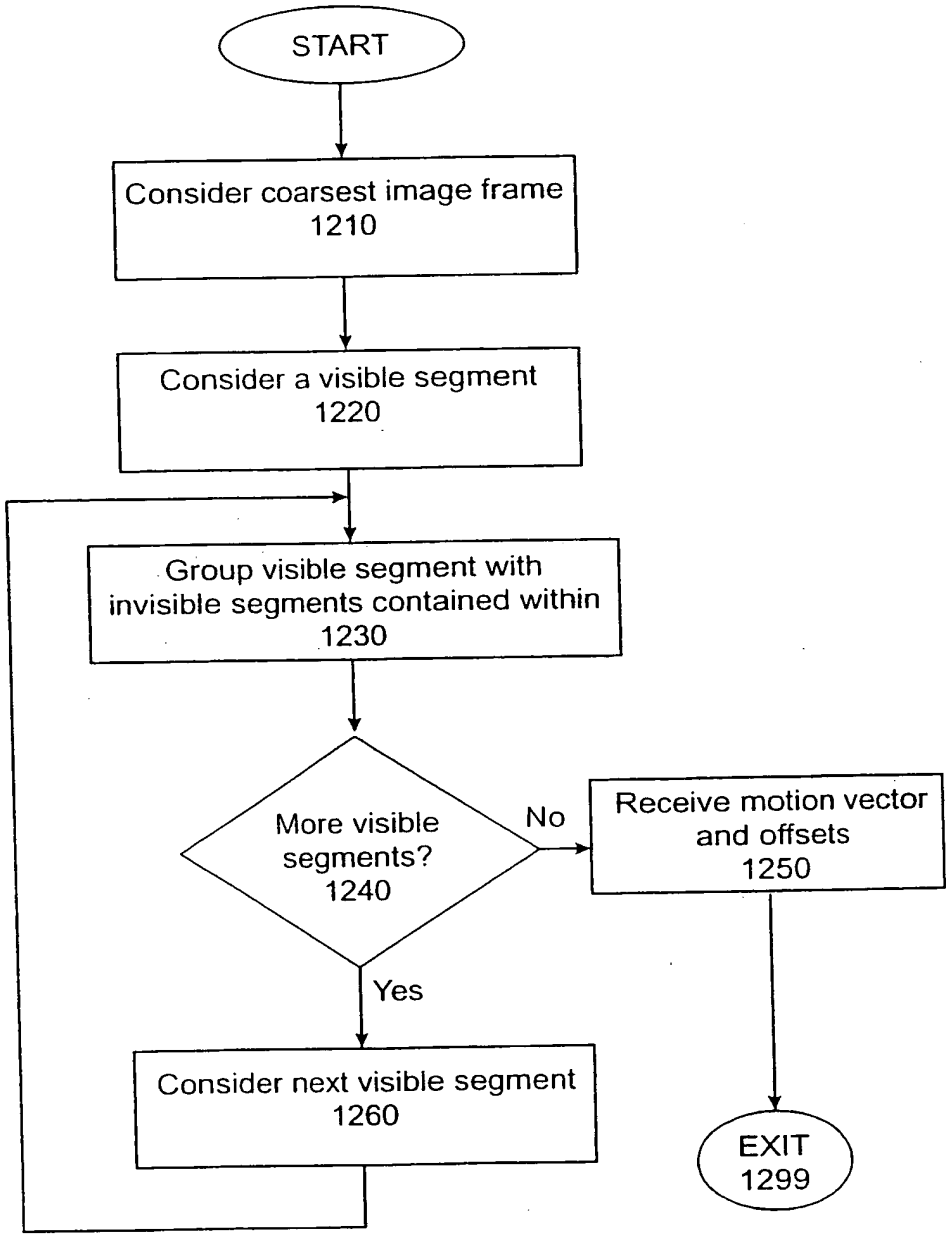
START
Consider coarsest image frame 1210
Consider a visible segment 1220
Group visible segment with invisible segments contained within 1230
More visible segments? 1240
No
Receive motion vector and offsets 1250
Yes
Consider next visible segment 1260
EXIT 1299

FIG. 19

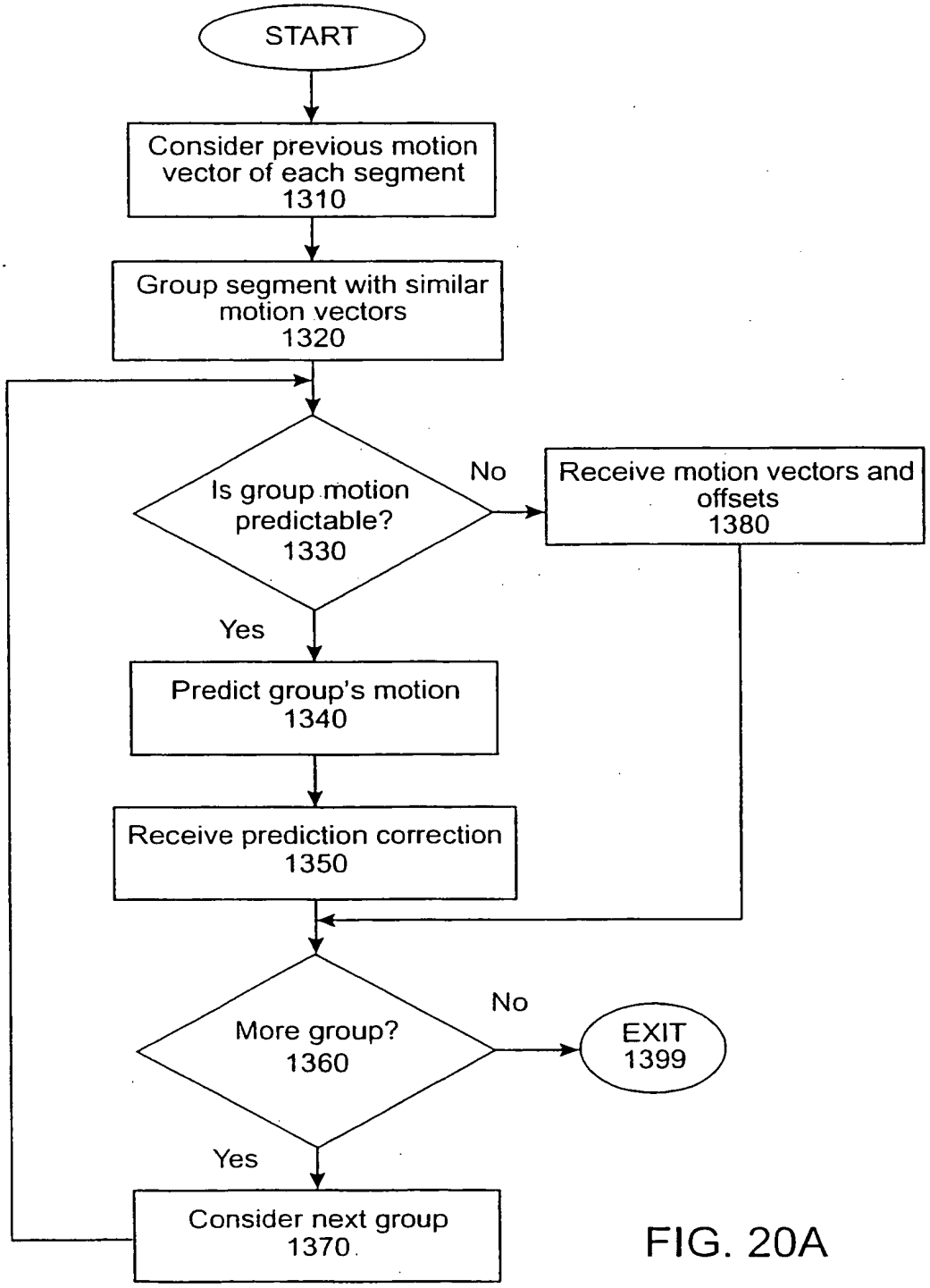
START
Consider previous motion vector of each segment 1310
Group segment with similar motion vectors 1320
Is group motion predictable? 1330
No
Receive motion vectors and offsets 1380
Yes
Predict group's motion 1340
Receive prediction correction 1350
More group? 1360
No
EXIT 1399
Yes
Consider next group 1370

FIG. 20A

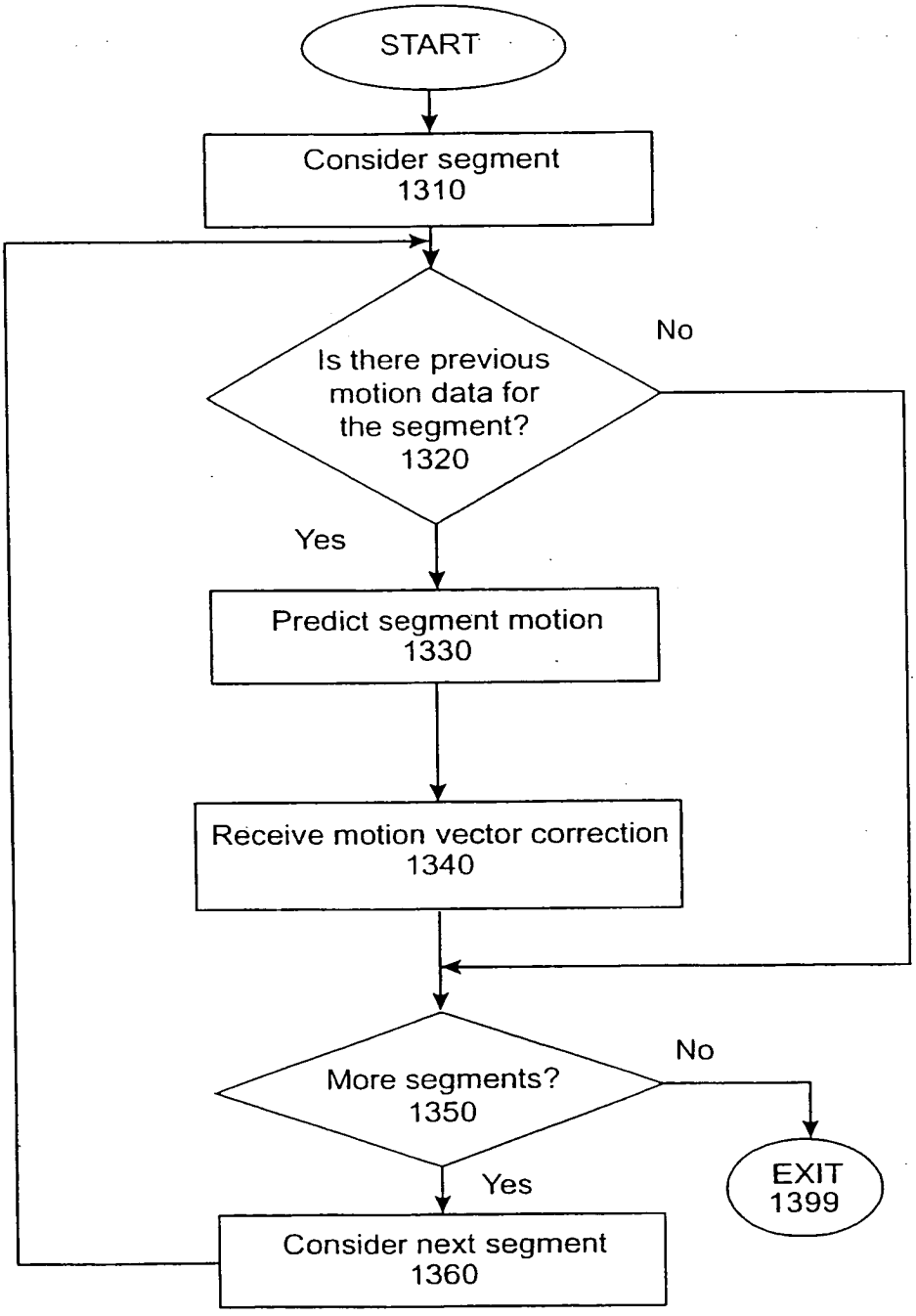
START
Consider segment
1310
Is there previous
motion data for
the segment?
1320
No
Yes
Predict segment motion
1330
Receive motion vector correction
1340
More segments?
1350
No
Yes
EXIT
1399
Consider next segment
1360

FIG. 20B

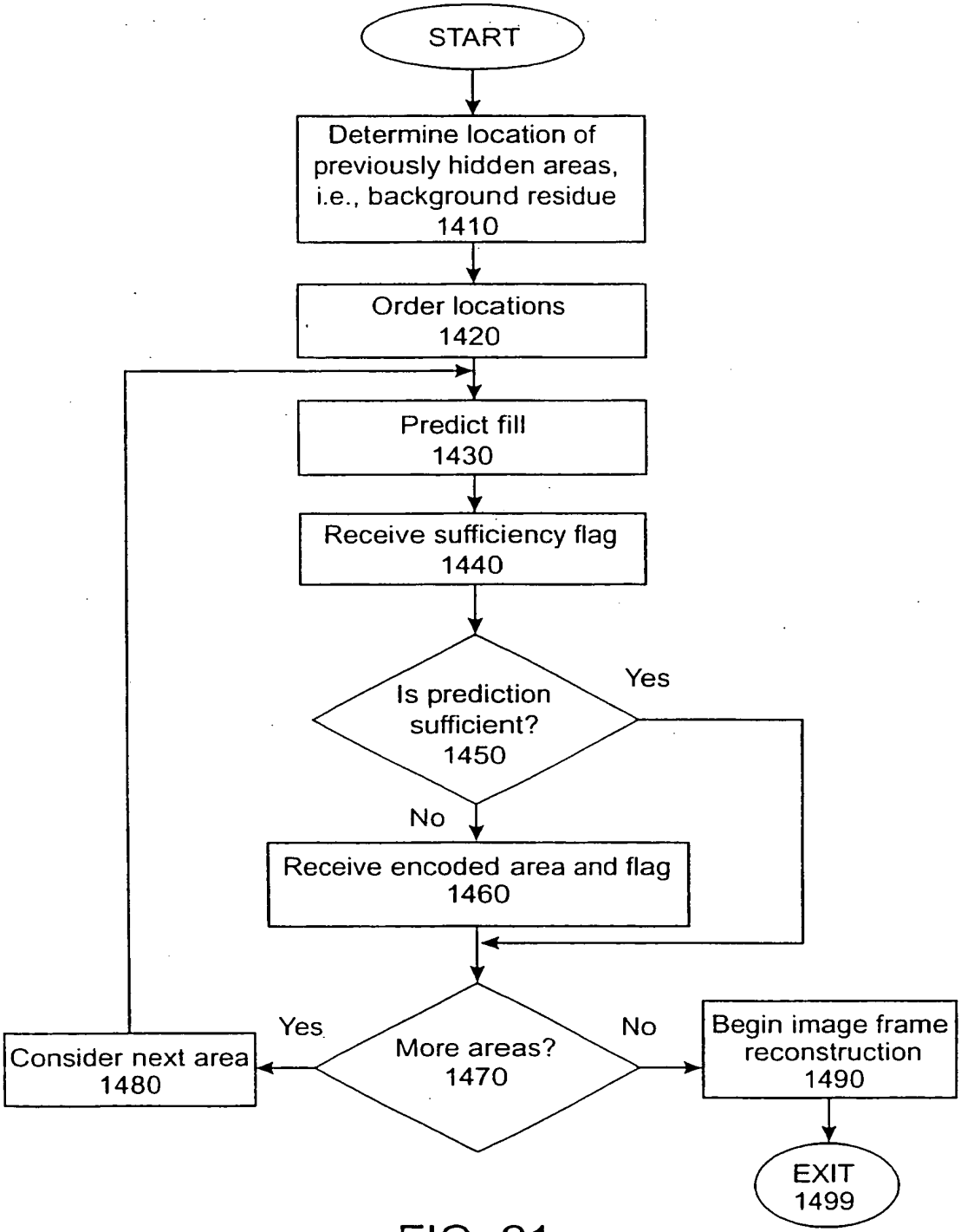
START
Determine location of previously hidden areas, i.e., background residue 1410
Order locations 1420
Predict fill 1430
Receive sufficiency flag 1440
Is prediction sufficient? 1450
Yes
No
Receive encoded area and flag 1460
More areas? 1470
Yes
No
Consider next area 1480
Begin image frame reconstruction 1490
EXIT 1499

FIG. 21

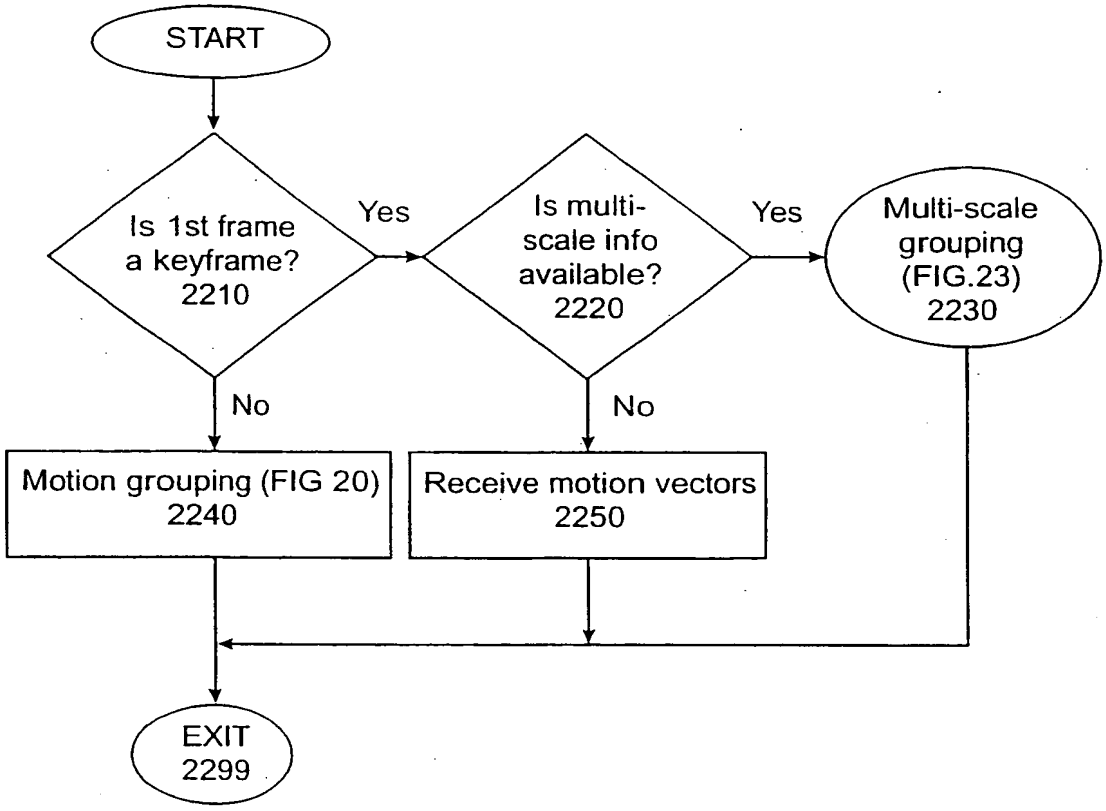
START
Is 1st frame a keyframe? 2210
Yes
Is multi-scale info available? 2220
Yes
Multi-scale grouping (FIG.23) 2230
No
No
Motion grouping (FIG 20) 2240
Receive motion vectors 2250
EXIT 2299

FIG. 22

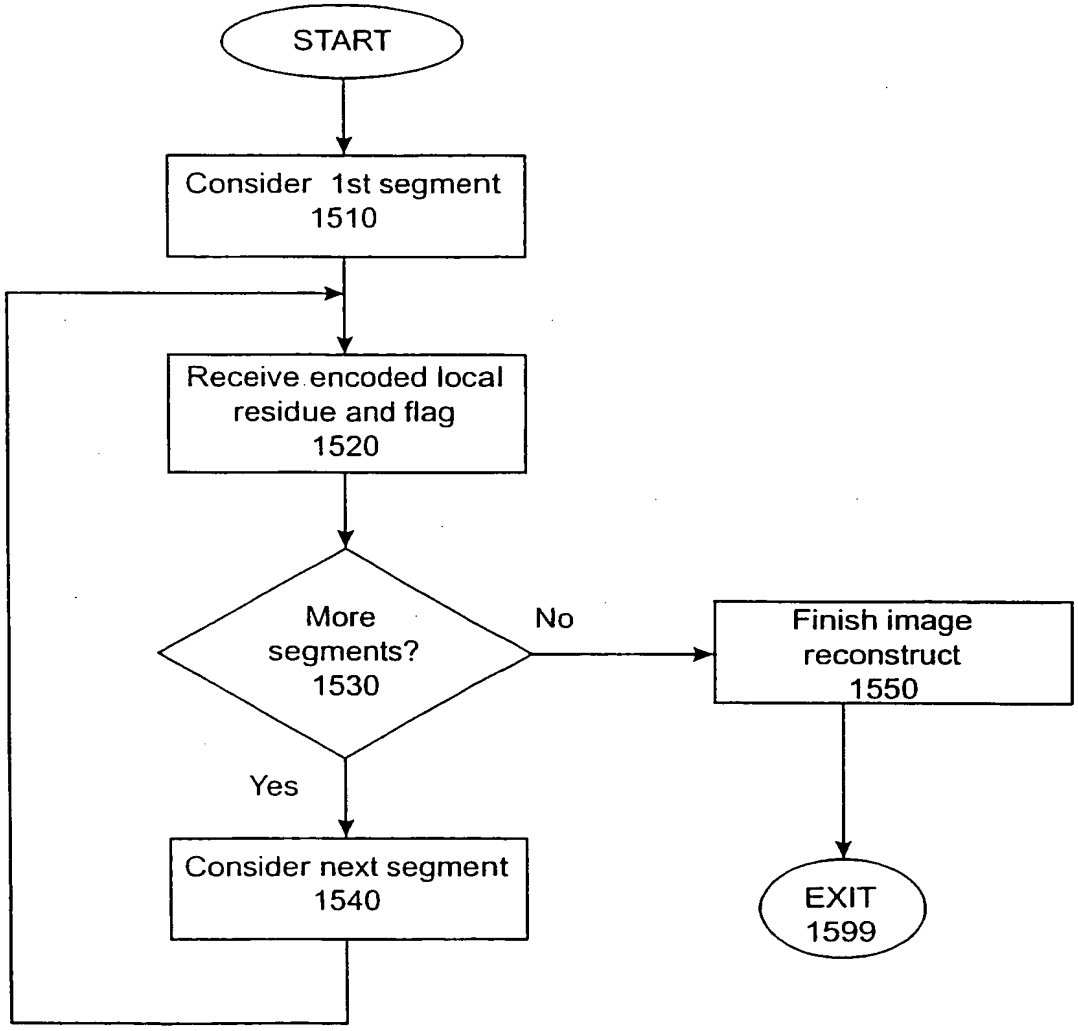
START
Consider 1st segment
1510
Receive encoded local residue and flag
1520
More segments?
1530
No
Finish image reconstruct
1550
Yes
Consider next segment
1540
EXIT
1599

FIG. 23

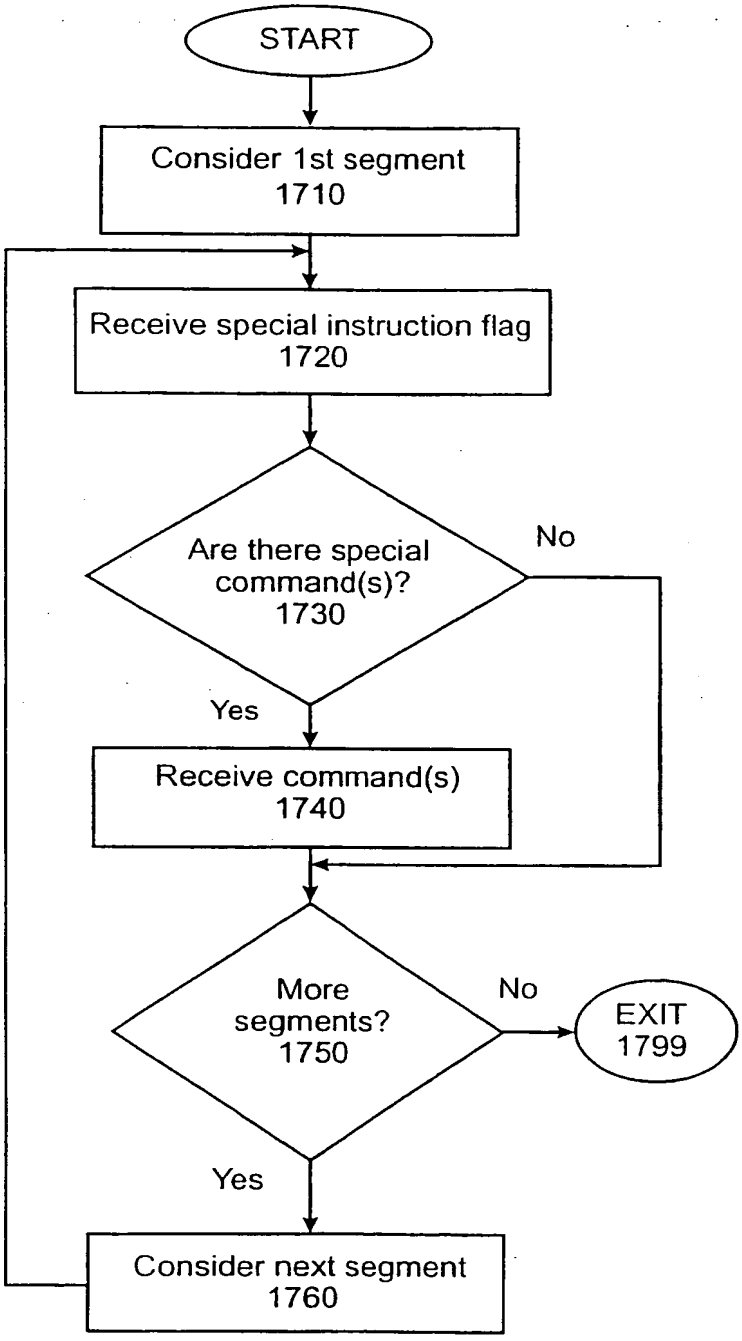
START
Consider 1st segment
1710
Receive special instruction flag
1720
Are there special
command(s)?
1730
No
Yes
Receive command(s)
1740
More
segments?
1750
No
EXIT
1799
Yes
Consider next segment
1760

FIG. 24

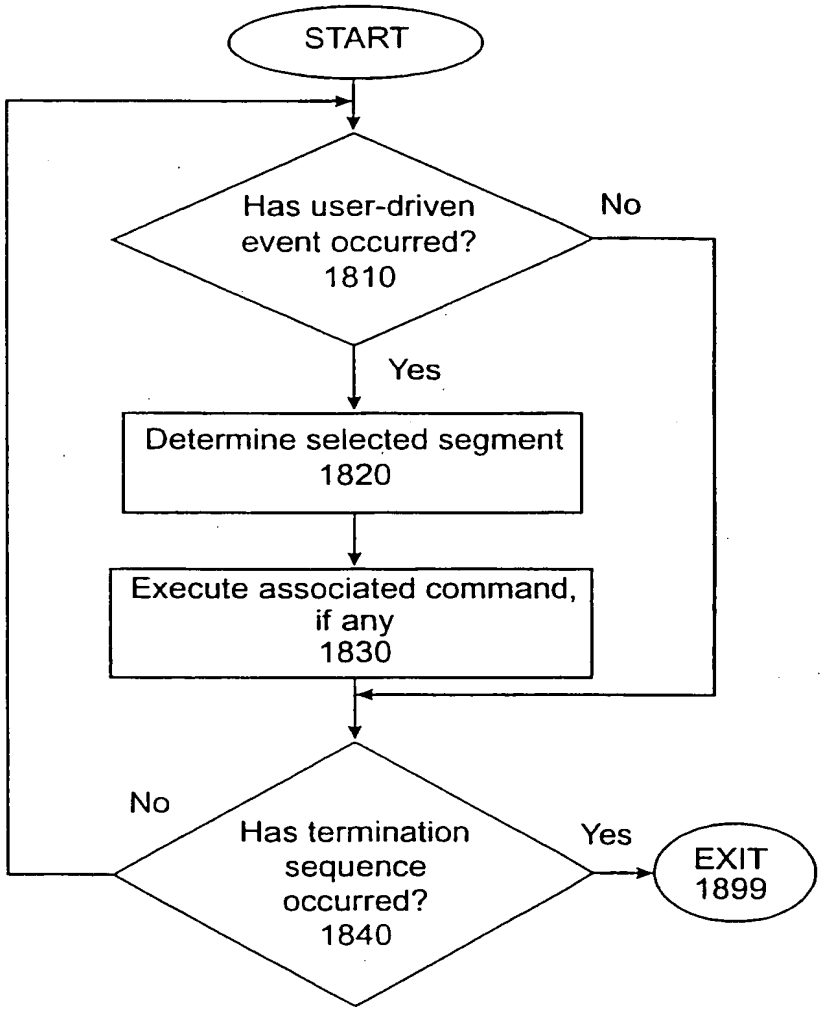
START
Has user-driven event occurred? 1810
No
Yes
Determine selected segment 1820
Execute associated command, if any 1830
Has termination sequence occurred? 1840
No
Yes
EXIT 1899

FIG.25

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US00/10451

A. CLASSIFICATION OF SUBJECT MATTER
IPC(7) :H04N 5/262
US CL :375/240
According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
U.S. : 375/240; 348/240

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X, P	US 6,026,182 A (LEE ET AL) 15 FEBRUARY 2000, col. 7, lines 43-52, col. 7, lines 63-67, col. 8, lines 1-5, col 10, lines 36-42, and Fig. 23b, element 706.	1.
X, E	US 6,057,884 A (CHEN ET AL) 02 MAY 2000, Fig. 3.	1.

☐ Further documents are listed in the continuation of Box C. ☐ See patent family annex.

*	Special categories of cited documents:	"T"	later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"A"	document defining the general state of the art which is not considered to be of particular relevance	"X"	document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"E"	earlier document published on or after the international filing date	"Y"	document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
"L"	document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)	"&"	document member of the same patent family
"O"	document referring to an oral disclosure, use, exhibition or other means		
"P"	document published prior to the international filing date but later than the priority date claimed		

Date of the actual completion of the international search 22 JUNE 2000	Date of mailing of the international search report 26 JUL 2000
Name and mailing address of the ISA/US Commissioner of Patents and Trademarks Box PCT Washington, D.C. 20231 Facsimile No. (703) 305-3230	Authorized officer CHRISTOPHER KELLEY James R. Matthews Telephone No. (703) 305-4856

Form PCT/ISA/210 (second sheet) (July 1998)*